



TUGAS AKHIR - KS141501

RANCANG BANGUN KLASTERISASI DATA CUACA DENGAN MENGGUNAKAN METODE HIERARCHICAL CLUSTERING

DEVELOPMENT WEATHER DATA CLUSTERING USING METHODS HIERARCHICAL CLUSTERING

**IZZATI AKAGA RIANMA ETWITRIEKA
NRP 5212 100 067**

**Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D
Irmasari Hafidz, S.kom, M.Sc**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

TUGAS AKHIR - KS141501

RANCANG BANGUN KLASTERISASI DATA CUACA DENGAN MENGGUNAKAN METODE HIERARCHICAL CLUSTERING

**IZZATI AKAGA RIANMA ETWITRIEKA
NRP 5212 100 067**

**Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D
Irmasari Hafidz, S.kom, M.Sc**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

FINAL PROJECT - KS 141501

DEVELOPMENT WEATHER DATA CLUSTERING USING METHODS HIERARCHICAL CLUSTERING

**IZZATI AKAGA RIANMA ETWITRIEKA
NRP 5212 100 067**

Supervisors

**Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D
Irmasari Hafidz, S.kom, M.Sc**

**INFORMATION SYSTEMS DEPARTMENT
Information Technology Faculty
Sepuluh Nopember Institut of Technology
Surabaya 2016**

LEMBAR PENGESAHAN

RANCANG BANGUN KLASTERISASI DATA CUACA DENGAN MENGGUNAKAN METODE HIERARCHICAL CLUSTERING TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

IZZATI AKAGA RIANMA ETWITRIEKA

NRP. 5212 100 067

Surabaya, Juli 2017

**KETUA
JURUSAN SISTEM INFORMASI**



Dr. Ir. Aris Tjahyanto, M.Kom.

NIP. 19650310 199102 1 001

LEMBAR PERSETUJUAN

RANCANG BANGUN KLASTERISASI DATA CUACA DENGAN MENGGUNAKAN METODE HIERATICAL CLUSTERING

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

IZZATI AKAGA RIANMA ETWITRIEKA

NRP. 5212 100 067

Disetujui Tim Penguji : Tanggal Ujian: 12 Juli 2017
Periode Wisuda:

Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

(Pembimbing I)

Irmasari Hafidz, S.kom., M.Sc.

(Pembimbing II)

Faizal Johan Atletiko, S.Kom., M.T.

(Penguji I)

Nisfu Asrul Sani S.Kom., M.Sc.

(Penguji II)

RANCANG BANGUN KLASTERISASI DATA CUACA DENGAN MENGGUNAKAN METODE HIERARCHICAL CLUSTERING

Nama Mahasiswa : IZZATI AKAGA RIANMA
ETWITRIEKA
NRP : 5212 100067
Jurusan : SISTEM INFORMASI
FTIF-ITS
Dosen Pembimbing 1 : Nur Aini Rakhmawati, S.Kom,
M.Sc.Eng, Ph.D
Dosen Pembimbing 2 : Irmasari Hafidz, S.Kom, M.Sc

ABSTRAK

Cuaca merupakan hal yang tak terpisahkan dengan manusia. Hampir sebagian besar aktivitas manusia bergantung dengan kondisi cuaca yang ada di sekitarnya. Walaupun aktivitas manusia banyak bergantung pada cuaca, manusia tidak dapat menilai kondisi cuaca dengan baik secara langsung. Seiring berkembangnya teknologi informasi, maka cara untuk mendapatkan informasi cuaca menjadi lebih mudah melalui aplikasi-aplikasi pada smartphone maupun pada personal computer atau melalui web yang menyediakan informasi cuaca. Akan tetapi permasalahan tidak benar-benar terselesaikan, karena lokasi sensor untuk mengukur kondisi cuaca sangat terbatas sehingga keakuratan pengukuran pada lokasi yang semakin jauh dari sensor akan semakin berkurang. Arduino sebagai sebuah microcontroller dapat menggantikan peran dari sensor tersebut. Selain harga yang murah, arduino tidak membutuhkan tempat yang luas dan mudah dipindahkan. Akan tetapi data yang dihasilkan oleh arduino masih berupa angka-angka sehingga manusia masih sulit untuk memahaminya. Oleh karena itu data dari arduino tersebut perlu di klasterisasi dan ditampilkan pada halaman web agar dapat diakses

oleh banyak orang. Proses klasterisasi akan menggunakan metode hierarchical clustering dan akan menghasilkan 5 klaster. Untuk membentuk hirarki menggunakan pendekatan agglomerative, sedangkan untuk menghitung jarak menggunakan rumus squared euclidean distance.

Kata kunci : Cuaca, Arduino, Hieratical Clustering

DEVELOPMENT WEATHER DATA CLUSTERING USING METHODS HIERARCHICAL CLUSTERING

**Name : IZZATI AKAGA RIANMA
ETWITRIEKA**
NRP : 5212 100067
**Departement : INFORMATION SYSTEM
FTIF-ITS**
**Supervisor 1 : Nur Aini Rakhmawati, S.Kom,
M.Sc.Eng, Ph.D**
Supervisor 2 : Irmasari Hafidz, S.Kom, M.Sc

ABSTRACT

Weather is inseparable from humans. Almost most human activities depend on the weather conditions around them. Although human activity depends a lot on the weather, humans can not properly assess the weather conditions directly. As the development of information technology, then the way to get the weather information becomes easier through applications on the smartphone or on personal computers or through the web that provides weather information. But the problem is not really resolved, because the location of the sensor to measure weather conditions is so limited that the accuracy of the measurement at the farther location of the sensor will decrease. Arduino as a microcontroller can replace the role of the sensor. In addition to the cheap price, the arduino does not need a large and easy to move place. But the data produced by arduino is still a number so that humans are still difficult to understand. Therefore the data from the arduino needs to be clustered and displayed on a web page to

be accessible to many people. Clustering process will use hierarchical clustering method and will produce 5 clusters. To form a hierarchy using the agglomerative approach, whereas to calculate the distance using the squared euclidean distance formula.

Keywords : Weather, Arduino, Hieratical Clustering

KATA PENGANTAR

Alhamdulillah atas karunia, rahmat, barakah, dan jalan yang telah diberikan Allah SWT selama ini memberikan kekuatan dan kehidupan kepada penulis dalam menyelesaikan tugas akhir dengan judul:

RANCANG BANGUN KLASTERISASI DATA CUACA DENGAN MENGGUNAKAN METODE HIERARCHICAL CLUSTERING

Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari beberapa pihak yang telah meluangkan waktu, tenaga dan pikirannya bagi penulis untuk menyelesaikan tugas akhir ini. Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak-pihak di bawah ini:

- Ibu Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D, selaku dosen pembimbing I yang telah melunaskan waktu dan pikiran beliau untuk membimbing dan mengarahkan penulis dalam pengerjaan tugas akhir ini.
- Ibu Irmasari Hafidz, S.Kom, M.Sc, selaku dosen pembimbing I yang telah melunaskan waktu dan pikiran beliau untuk membimbing dan mengarahkan penulis dalam pengerjaan tugas akhir ini.
- Orang tua penulis yang senantiasa memberi dukungan moral kepada penulis sehingga dapat mengerjakan tugas akhir ini.
- Teman-teman seperjuangan yang telah senantiasa pengerjaan tugas akhir ini.
- Serta seluruh pihak yang telah membantu penulis dalam mengerjakan tugas akhir ini yang tidak mungkin disebut satu per satu.

Penulis pun menyadari bahwa tugas akhir ini masih belum sempurna dengan segala kekurangan di dalamnya. Oleh Karena itu penulis memohon maaf atas segala kekurangan dan kekeliruan yang ada di dalam tugas akhir ini. Penulis membuka pintu

selebar-lebarnya bagi pihak-pihak yang ingin memberikan kritik dan saran bagi penulis untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi seluruh pembaca

Surabaya, Juli 2017

Penulis

DAFTAR ISI

ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Tugas Akhir	3
1.5. Manfaat Tugas Akhir	3
1.6. Relevansi	4
BAB II TINJAUAN PUSTAKA	5
2.1. Studi Sebelumnya	5
2.2. Dasar Teori	6
2.2.1. Arduino	6
2.2.2. Hierarchical Clustering	6
2.2.3. Meteorologi	10
BAB III METODOLOGI PENELITIAN	13
3.1. Studi literatur	13
3.2. Perancangan Perangkat Lunak	13
3.3. Pembangunan Perangkat Lunak	13
3.3.1. Prototipe Pertama	13
3.3.2. Prototipe Kedua	14
3.3.3. Prototipe Ketiga	14
BAB IV ANALISA DAN DESAIN SISTEM	17
4.1. Proses Bisnis	17
4.2. Kebutuhan Fungsional	17
4.3. Kebutuhan Non-fungsional	18
4.4. Desain Tampilan Antarmuka	18
4.4.1. Tampilan Data Sekarang	20
4.4.2. Tampilan Hierarchical Tree	21
4.4.3. Tampilan Data Lampau	22
4.4.4. Tampilan Unduh	23
4.5. Desain Database	24

4.6. Test Case.....	25
BAB V IMPLEMENTASI	29
5.1. Lingkungan Implementasi	29
5.1.1. Lingkungan Implementasi Perangkat Keras	29
5.1.2. Lingkungan Implementasi Perangkat Lunak	29
5.2. Kode Perangkat Lunak.....	29
5.2.1. Kode Pada Database	29
5.2.2. Kode Pada Website.....	36
BAB VI PENGUJIAN DAN PEMBAHASAN	41
6.1. Pengujian Kecepatan Klasterisasi	41
6.2. Pengujian Keakuratan Klasterisasi	41
6.3. Pengujian Fungsi.....	44
6.4. Pembahasan	45
BAB VII KESIMPULAN DAN SARAN	47
7.1. Kesimpulan	47
7.2. Saran	47
DAFTAR PUSTAKA	49
LAMPIRAN	1
Source Code MySQL Event Membuat Rata-Rata Setiap 10 Menit.....	1
Source Code MySQL Procedure Klasterisasi	2
Source Code MySQL Event Memulai Proses Klasterisasi Setiap Minggu	7
Source Code PHP Untuk Mengakses Tabel ‘node’	8
Source Code PHP Untuk Mengakses Tabel ‘rekap’	9
Source Code PHP Untuk Mengakses Tabel ‘sensor’ .	10
Source Code PHP Untuk Mengakses Tabel ‘tengah’	12
Source Code PHP Untuk Mengakses Tabel ‘tree’	15
Source Code PHP Controller Main.....	16
Source Code PHP Controller Unduh	26
Source Code PHP, HTML, dan Java Script Tampilan Inti.....	27
Source Code PHP, HTML, dan Java Script Untuk Membuat Bagan Balok	33
Source Code PHP, HTML, dan Java Script Untuk Membuat Bagan Pohon.....	39

DAFAR GAMBAR

Gambar 2.1 Contoh Data.....	8
Gambar 2.2 Jarak Antar Klater 1.....	8
Gambar 2.3 Hierarki 1.....	9
Gambar 2.4 Jarak Antar Klaster 2.....	9
Gambar 2.5 Hierarki 2.....	9
Gambar 2.6 Jarak Antar Klaster 3.....	9
Gambar 2.7 Hierarki 3.....	10
Gambar 2.8 Hierarki 4.....	10
Gambar 3.1 Alur Penelitian.....	15
Gambar 4.1 Desain Tampilan 1.....	21
Gambar 4.2 Desain Tampilan 2.....	22
Gambar 4.3 Desain Tampilan 3.....	23
Gambar 4.4 Desain Tampilan 4.....	24
Gambar 4.5 Desain Database	25
Gambar 6.1 Hasil Klasterisasi	42

DAFTAR TABEL

Tabel 4.1 Fitur Desain Tampilan Antarmuka.....	18
Tabel 4.2 Test Case	25
Tabel 6.1 Perbandingan Kecepatan Klasterisasi	41
Tabel 6.2 Data Dummy	41
Tabel 6.3 Perhitungan Jarak 1	42
Tabel 6.4 Node 1	43
Tabel 6.5 Perhitungan Jarak 2	43
Tabel 6.6 Node 2	43
Tabel 6.7 Perhitungan Jarak 3	44
Tabel 6.8 Node 3	44
Tabel 6.9 Node 4	44
Tabel 6.10 Keluaran Prototipe Ketiga	44

BAB I

PENDAHULUAN

Pada bab pendahuluan akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

1.1. Latar Belakang

Aktifitas manusia tidak akan terlepas dari kondisi cuaca yang ada disekitarnya terutama aktifitas di luar ruangan. Karena banyak aktivitas manusia yang berkaitan dengan cuaca di sekitarnya maka manusia perlu mengetahui kondisi cuaca yang ada di sekitarnya apakah memungkinkan untuk melakukan aktivitas atau tidak. Akan tetapi untuk dapat mengetahui kondisi cuaca di sekitar kita bukanlah hal yang mudah karena manusia tidak dapat mengukur indikator-indikator cuaca secara langsung. Selain tidak dapat mengukur indikator-indikator cuaca secara langsung manusia juga perlu mengetahui kondisi indikator-indikator cuaca pada waktu yang telah lalu untuk dapat membandingkan dan melakukan penilaian tentang kondisi cuaca sekarang.

Semakin berkembangnya teknologi informasi, saat ini sudah banyak teknologi informasi yang dapat membantu manusia untuk mendapatkan informasi tentang kondisi cuaca disekitarnya. Teknologi informasi tersebut juga mudah diakses dari manapun karena sudah tersedia dalam bentuk aplikasi pada *smartphone* maupun pada bentuk halaman web. Selain dapat menampilkan kondisi cuaca pada saat ini teknologi informasi tersebut juga dapat menampilkan perkiraan kondisi cuaca untuk waktu yang akan datang. Walaupun sudah memiliki banyak keunggulan tetapi teknologi informasi tersebut juga memiliki

kelemahan. Salah satu kelemahan yang utama adalah alat yang digunakan untuk mengukur kondisi cuaca hanya ditempatkan di beberapa titik di suatu kota. Karena hanya ditempatkan di beberapa titik saja maka informasi kondisi cuaca hanya akurat pada lokasi yang dekat dengan lokasi sensor saja. penempatan sensor yang sedikit disebabkan oleh harga alat yang mahal serta membutuhkan tempat yang luas untuk melakukan pengukuran kondisi cuaca tersebut, serta alat tersebut juga sulit untuk dipindahkan.

Arduino merupakan sebuah microcontroller yang dapat dipasang beberapa sensor untuk mengukur indikator-indikator cuaca yang ada disekitar, sehingga dapat dimanfaatkan untuk membantu manusia memahami kondisi cuaca yang ada disekitarnya. Jika dibandingkan dengan alat pengukur kondisi cuaca yang sudah ada, arduino memiliki beberapa keunggulan yaitu harganya yang murah, tempat yang dibutuhkan untuk melakukan pengukuran yang tidak terlalu luas dan mudah untuk dipindahkan. Akan tetapi arduino saja belum cukup untuk dapat menyelesaikan masalah manusia dalam mengetahui kondisi cuaca yang ada disekitarnya, karena hanya dapat menampilkan pengukuran indikator cuaca pada saat ini. untuk itu diperlukan sebuah server yang dapat menampung data pengukuran indikator-indikator cuaca.

Dengan adanya server maka permasalahan yang tersisa adalah bagaimana cara membandingkan kondisi sekarang dan kondisi yang telah lalu. Setelah dibandingkan kita juga harus memberikan penilaian tentang kondisi cuaca sekarang dan perbedaannya dengan kondisi cuaca pada waktu lampau. Untuk itu salah satu solusinya adalah dengan mengelompokkan data hasil pengukuran indikator kondisi cuaca agar kita mengetahui gambaran kondisi cuaca sekarang seperti apa, dengan membandingkan kondisi cuaca waktu lampau. Untuk dapat mengelompokkan data-data pengukuran indikator cuaca di perlukan proses klasterisasi.

Setelah dilakukan proses klusterisasi maka hasil klusterisasi tersebut perlu ditampilkan dengan bentuk yang mudah dilihat oleh manusia. Media untuk menampilkan hasil klusterisasi dari data yang diukur oleh arduino haruslah mudah diakses dimana saja, untuk itu data tersebut akan ditampilkan ke dalam bentuk halaman web. Sehingga setelah dilakukan klusterisasi maka data tersebut akan ditampilkan ke dalam halaman web, agar mudah diakses oleh masyarakat.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang, maka rumusan permasalahan yang menjadifokus dan akan diselesaikan dalam Tugas Akhir ini antara lain:

1. Bagaimana melakukan klusterisasi data cuaca dari Arduino dengan menggunakan metode *Hierarchical Clustering*.
2. Bagaimana data hasil hierarchical clustering data cuaca dapat disajikan pada sebuah web.

1.3. Batasan Masalah

Dari permasalahan yang disebutkan di atas, batasan masalah dalam tugasakhir ini adalah :

1. Arduino yang digunakan untuk pengumpulan data hanya dipasang pada 1 titik di wilayah ITS.
2. Tugas akhir ini tidak membahas pemasangan dan perangkaian sensor-sensor pada Arduino yang akan digunakan untuk pengambilan data.

1.4. Tujuan Tugas Akhir

Tujuan dari tugas akhir ini adalah untuk mencoba melakukan klusterisasi dengan metode *hierarchical clustering* terhadap data yang didapatkan oleh Arduino.

1.5. Manfaat Tugas Akhir

Manfaat yang diperoleh dari tugas akhir ini adalah dapat menggambarkan kondisi cuaca terbaru yang diperoleh dari proses pengukuran menggunakan Arduino.

1.6. Relevansi

Tugas akhir ini berkaitan dengan mata kuliah Penggalian dan Analitika Bisnis dan Lab Akuisisi Data dan Diseminasi Informasi, karena tugas akhir ini bertujuan untuk melakukan klusterisasi data yang diambil melalui sensor yang dipasang pada arduino dan menyebarkan informasi tersebut melalui halaman web.

BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir ini.

2.1. Studi Sebelumnya

Berikut ini adalah daftar penelitian yang telah dilakukan sebelumnya yang mendasari penelitian project ini:

1. Jess Christopher B. Lopez, Harreez M. Villaruz "Low-Cost Weather Monitoring System with Online Logging and Data Visualization". Pada penelitian kali ini, penulis melakukan penelitian membuat sebuah sensor cuaca hanya sebatas menggunakan arduino, jadi jika terdapat sebuah perubahan cuaca, Arduino akan menampilkan perubahan cuaca tersebut pada bagian monitor [1].
2. Penelitian Cholatip Yawut and Sathapath Kilaso (2011) mengenai A Wireless Sensor Network for Weather and Disaster Alarm Systems: pada penelitian ini penulis membuat sebuah pemantau cuaca yang terkoneksi dengan sebuah jaringan nirkabel, sehingga jika terjadi kondisi ekstrim yang ada pada lokasi sensor, maka alarm yang ada akan berbunyi untuk memberi peringatan kepada pemilik [2].
3. Rajendra Kumar Roul, Shubham Rohan Asthana, Sanjay Kumar Sahay Automated Document Indexing via Intelligent Hierarchical Clustering: A Novel Approach. Pada penelitian kali ini penulis menggunakan metode *Hierarchical Clustering* untuk melakukan indeksasi terhadap buku digital secara otomatis [3].

2.2. Dasar Teori

2.2.1. Arduino

Arduino merupakan suatu perusahaan perangkat lunak, komunitas pengguna dan proyek. Fokus utama dari arduino adalah merancang dan memproduksi perangkat keras, perangkat lunak dan perangkat *microcontroller* yang bersifat open source yang dapat menangkap ransangan dan mengendalikan perangkat fisik. Selain bersifat open source keunggulan lain dari arduino adalah arduino dapat digunakan pada berbagai sistem operasi seperti Macintosh OSX, Windows dan Linux melalui Arduino Software (IDE). Untuk bahasa pemrograman yang digunakan pada arduino mirip dengan bahasa pemrograman C++.

2.2.2. Hierarchical Clustering

Hierarchical clustering merupakan salah satu metode yang digunakan untuk melakukan klasterisasi. Klasterisasi sendiri adalah sebuah proses yang dilakukan dengan tujuan untuk mengelompokkan data yang memiliki karakteristik yang sama pada suatu kelompok dan data yang memiliki karakteristik yang berbeda pada kelompok yang lain. *Hierarchical clustering* adalah metode yang membentuk hierarki berdasarkan kemiripan suatu data dengan data yang lain. [4]

Untuk membuat hierarki data dapat menggunakan 2 cara yaitu *Agglomerative* dan *Divisive*. *Agglomerative* menggunakan pendekatan *bottom-up* yaitu dengan mencari data mana yang memiliki karakteristik paling mirip dan menjadikannya satu klaster kemudian mencari klaster lain yang memiliki karakteristik paling mirip dan menggabungkan klaster tersebut. Proses ini kan terus diulang sampai di dapat jumlah klaster yang diinginkan. *Divisive* menggunakan pendekatan *top-down* yaitu dengan cara membuat satu klaster yang

mencakup semua data kemudian dicari yang memiliki karakteristik yang paling tidak mirip dan mengeluarkannya dari klaster dan membuat klaster baru. Proses itu diulang sampai didapatkan jumlah klaster yang diinginkan.

Untuk dapat menentukan kemiripan suatu data dapat menggunakan menggunakan beberapa rumus seperti berikut:

- Euclidean distance

$$\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

- Squared Euclidean distance

$$\|a - b\|_2^2 = \sum_i (a_i - b_i)^2$$

- Manhattan distance

$$\|a - b\|_1 = \sum_i |a_i - b_i|$$

- Maximum distance

$$\|a - b\|_\infty = \max_i |a_i - b_i|$$

Sebagai contoh, jika diketahui data seperti Gambar 2.1 Contoh Data. kita akan melakukan klasterisasi dengan menggunakan pendekatan *agglomerative*.

Nama	Temperatur (°C)
A	25
B	26
C	23
D	16
E	23

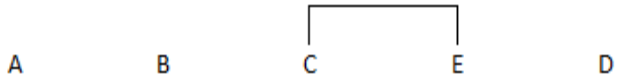
Gambar 2.1 Contoh Data

Langkah pertama kita membuat klaster sebanyak jumlah data yang ada dan setiap klaster hanya diisi satu data, selanjutnya kita akan mencari jarak antara setiap klaster yang ada. Dengan menggunakan euclidean distance maka diketahui jarak antara klaster satu dengan yang lainnya seperti pada Gambar 2.2 Jarak Antar Klater 1.

Kluster	A	B	C	D	E
A	-	1	2	9	2
B	1	-	3	10	3
C	2	3	-	7	0
D	9	10	7	-	7
E	2	3	0	7	-

Gambar 2.2 Jarak Antar Klater 1

Kemudian kita cari jarak yang paling sedikit, dan diketahui klaster C dan E memiliki jarak yang paling sedikit sehingga kita gabungkan menjadi 1 klaster seperti pada Gambar 2.3 Hierarki 1. Kemudian kita cari titik tengah kluter C-E, yaitu $(23 + 23) / 2 = 23$.



Gambar 2.3 Hierarki 1

Kemudian kita hitung kembali jarak antara kluster. Seperti yang diperlihatkan pada Gambar 2.4 Jarak Antar Kluster 2, bahwa jarak yang paling dekat adalah kluster A dan B.

Kluster	A	B	C-E	D
A	-	1	2	9
B	1	-	3	10
C-E	2	3	-	7
D	9	10	7	-

Gambar 2.4 Jarak Antar Kluster 2

Kemudian kita gabungkan kluster A dan B menjadi kluster A-B seperti pada Gambar 2.5 Hierarki 2, dengan titik tengah kluster adalah $(25 + 26) / 2 = 25,5$.



Gambar 2.5 Hierarki 2

Kemudian kita hitung kembali jarak antara kluster. Seperti yang diperlihatkan pada Gambar 2.6 Jarak Antar Kluster 3, bahwa jarak yang paling dekat adalah kluster A-B dan C-E.

Kluster	A-B	C-E	D
A-B	-	2,5	9,5
C-E	2,5	-	7
D	9,5	7	-

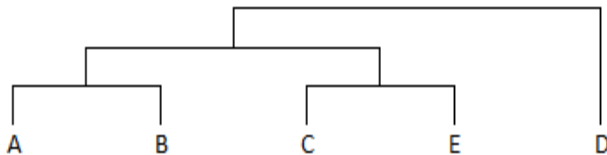
Gambar 2.6 Jarak Antar Kluster 3

Kemudian kita gabungkan kluster A-B dan C-E menjadi kluster A-B-C-E seperti pada Gambar 2.7 Hierarki 3 dengan titik tengah kluster adalah $(25,5 + 23) / 2 = 24,25$.



Gambar 2.7 Hierarki 3

Karena sekarang hanya tersisa 2 kluster saja yaitu kluster A-B-C-E dan D maka otomatis kedua kluster tersebut menjadi kluster yang paling dekat maka kita gabung dua kluster tersebut menjadi kluster A-B-C-D-E seperti Gambar 2.8 Hierarki 4 dengan titik tengah $(24,25 + 16) / 2 = 20,125$, sehingga jika kita menginginkan 2 kluster maka ada kluster A-B-C-E dan kluster D, dan jika 3 kluster maka ada kluster A-B, C-E dan D.



Gambar 2.8 Hierarki 4

Metode *hierarchical clustering* memiliki keunggulan utama yaitu tidak perlu menentukan jumlah kluster, karena kita bisa memilih jumlah kluster dari hierarki yang telah dihasilkan. Selain itu jika ada data baru yang masuk dari maka hanya perlu mencari data yang terdekat dengan data baru dan menggabungkan data tersebut pada hierarki yang sudah ada\cite{clustering}. Dari segi kecepatan dalam membuat suatu kluster, algoritma *hierarchical clustering* lebih lambat dari algoritma *K-means* akan tetapi dari segi ketepatan algoritma *hierarchical clustering* lebih baik. [5]

2.2.3. Meteorologi

Meteorologi merupakan suatu ilmu yang mempelajari tentang keadaan cuaca [6]. Cara untuk mempelajari

kondisi cuaca adalah dengan mengamati indikator-indikator cuaca yang terdapat pada lapisan troposfer bumi atau pada ketinggian 0-10 kilometer dari atas permukaan laut yang merupakan lapisan atmosfer yang paling bawah. Indikator-indikator cuaca yang biasanya diukur adalah:

- **Kelembapan Udara**
Kelembapan udara adalah jumlah kandungan uap air yang ada dalam udara. Kandungan uap air berubah ubah tergantung suhu udara pada wilayah tersebut, sehingga kelembapan udara juga tergantung pada suhu udara.
- **Kecepatan Angin**
Kecepatan angin adalah kecepatan perpindahan udara dari suatu wilayah. Seperti yang kita ketahui bahwa udara berpindah dari tempat yang dingin ke tempat yang panas maka kecepatan angin juga dipengaruhi oleh suhu suatu wilayah.
- **Suhu Udara**
Suhu udara adalah derajat panas dan dingin udara di wilayah tersebut. Suhu udara di permukaan bumi dipengaruhi oleh faktor intensitas cahaya matahari. Hal itu dapat berdampak langsung akan adanya perubahan suhu di udara.
- **Intensitas Cahaya**
Intensitas cahaya matahari adalah kualitas cahaya matahari menyinari suatu wilayah. intensitas cahaya matahari ditentukan oleh beberapa faktor, yaitu sudut datang sinar matahari, lama waktu penyinaran, topografi, komposisi udara, keadaan tanah dan sifat permukaan bumi.

(Halaman ini sengaja dikosongkan)

BAB III METODOLOGI PENELITIAN

Pada bab metode penelitian akan dijelaskan mengenai tahapan – tahapan apa saja yang dilakukan dalam pengerjaan tugas akhir ini beserta deskripsi dan penjelasan tiap tahapan tersebut. Secara garis besar alur pengerjaan tugas akhir dijelaskan pada Gambar 3.1 Alur Penelitian.

3.1. Studi literatur

Pada tahap ini dilakukan pengumpulan literatur yang mendukung dalam menyelesaikan tugas akhir ini. Literatur disini adalah penjelasan konsep – konsep atau penelitian sebelumnya yang pernah dilakukan dan didokumentasikan dalam buku, jurnal, maupun website. Output atau keluaran proses ini adalah pemahaman tentang cara melakukan klasterisasi metode *hierarchical clustering* dan daftar kebutuhan perangkat lunak dari halaman web yang akan dibuat.

3.2. Perancangan Perangkat Lunak

Pada tahap ini akan dilakukan analisa dan desain untuk menampilkan data hasil klasterisasi pada halaman web. Pada tahap ini akan dihasilkan desain *user interface* untuk menampilkan hasil pengukuran dari Arduino dan *test case* yang akan digunakan untuk *blackbox testing*.

3.3. Pembangunan Perangkat Lunak

Setelah tahap perancangan perangkat lunak, maka akan dilanjutkan dengan tahap pembangunan perangkat lunak. pembangunan perangkat lunak akan mengacu pada metode *prototyping* [7] yang dimana akan di buat 3 buah prototipe dari halaman web ini.

3.3.1. Prototipe Pertama

Prototipe pertama adalah basis data yang dapat melakukan proses klasterisasi secara otomatis dan rutin. Proses klasterisasi menggunakan pendekatan *Agglomerative* yang dimana perhitungan jarak

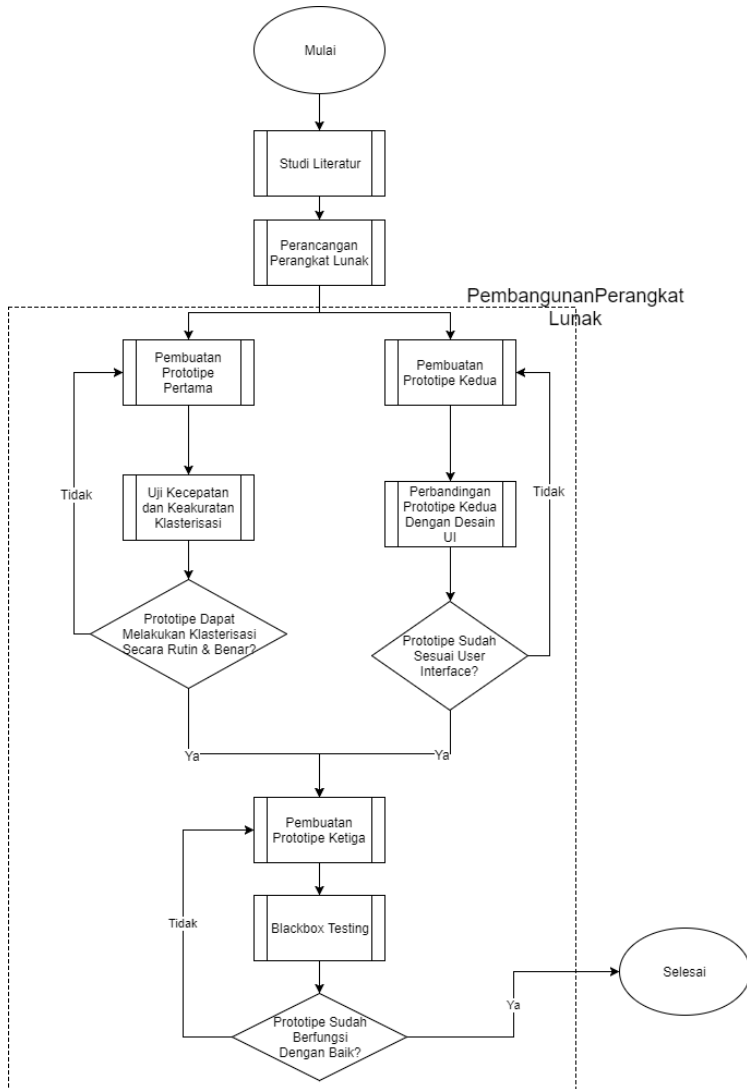
dilakukan dengan menggunakan rumus *Squared Euclidean distance*. Prototipe pertama memiliki target agar sistem dapat melakukan klasterisasi secara rutin dan otomatis. Pengujian prototipe pertama bertujuan untuk memastikan apakah klaster yang dibuat telah sesuai dan apakah proses klasterisasi dapat berjalan secara rutin. Pengujian pada prototipe ini adalah dengan melakukan pengujian pada kecepatan dan keakurasian dari hasil klasterisasi.

3.3.2. Prototipe Kedua

Prototipe kedua adalah halaman web yang memiliki tampilan mirip dengan desain tampilan antarmuka. Prototipe kedua memiliki target berupa tampilan halaman web yang sesuai dengan rancangan. Pada prototipe kedua ini halaman web tidak diharuskan menampilkan data yang sesuai atau memiliki fungsi yang berjalan dengan baik. Pengujian dari prototipe kedua adalah dengan membandingkan desain *user interface* dan tampilan halaman web.

3.3.3. Prototipe Ketiga

Prototipe ketiga merupakan prototipe terakhir dan gabungan dari prototipe pertama dan prototipe kedua. Target dari prototipe ketiga adalah halaman web yang sudah memiliki fungsi yang dibutuhkan dan dapat menampilkan data yang sesuai. Pengujian dari prototipe ketiga adalah dengan melakukan *blackbox testing* sesuai *test case* yang telah dibuat.

**Gambar 3.1 Alur Penelitian**

(Halaman ini sengaja dikosongkan)

BAB IV

ANALISA DAN DESAIN SISTEM

Pada bab empat ini akan membahas Analisa dan desain system yang akan dibuat.

4.1. Proses Bisnis

Website yang akan dibuat harus dapat menampilkan data hasil pengukuran Arduino terhadap indikator-indikator kondisi keadaan cuaca saat ini. Data yang ditampilkan akan diperbaharui setiap 10 detik sekali sesuai dengan waktu yang dibutuhkan Arduino untuk melakukan pengukuran. Data hasil pengukuran saat ini akan di bandingkan dengan data hasil klasterisasi data hasil pengukuran minggu lalu untuk dapat di perkirakan data hasil pengukuran saat ini akan masuk klaster yang mana. Hasil Klasterisasi menggunakan hierarchical clustering akan dibagi menjadi 5 klaster. Setiap 10 menit sekali akan dihitung rata-rata dari setiap nilai pengukuran indikator kondisi cuaca untuk didapatkan data rata-rata, data rata-rata inilah yang akan digunakan dalam proses klasterisasi setiap minggunya.

4.2. Kebutuhan Fungsional

Kebutuhan Fungsional merupakan kebutuhan perilaku yang harus ada pada sistem. Kebutuhan ini di dasarkan pada proses bisnis yang telah dijelaskan di awal. Kebutuhan fungsional dari website ini meliputi:

- Website dapat menampilkan nilai indikator kondisi keadaan cuaca saat ini.
- Website dapat menampilkan kluster terdekat dengan data kondisi saat ini.
- Website dapat menampilkan perbandingan antara data saat ini dan klaster terdekat.
- Website dapat menampilkan Hierarchical Tree dari yang telah dibuat.

- Website dapat menampilkan hasil rekapitulasi klasterisasi indikator kondisi keadaan cuaca.
- Website dapat memberikan pengguna data hasil pengukuran indikator kondisi cuaca.
- Sistem dapat melakukan klasterisasi data secara rutin.

4.3. Kebutuhan Non-fungsional

Kebutuhan Non-fungsional merupakan kebutuhan yang harus dicapai agar mendukung pemenuhan kebutuhan fungsional, kebutuhan non-fungsional pada website ini meliputi:

- Semua fitur yang ada pada website dapat digunakan sebagai mana fungsinya.
- Tidak adanya menu atau tombol yang membingungkan pengguna.

4.4. Desain Tampilan Antarmuka

Desain tampilan antarmuka menjelaskan mengenai gambaran tampilan yang akan dilihat oleh pengguna yang secara garis besar tampilan website ini dibagi menjadi 4 bagian yaitu Data Sekarang, Hierarchical Tree, Data Lampau dan Unduh. Desain tampilan tersebut memiliki fitur-fitur yang memenuhi kebutuhan fungsional dari website ini seperti yang ditampilkan pada Tabel 4.1 Fitur Desain Tampilan Antarmuka. Desain tampilan antar muka inilah yang akan digunakan untuk melakukan pengujian pada prototipe kedua.

Tabel 4.1 Fitur Desain Tampilan Antarmuka

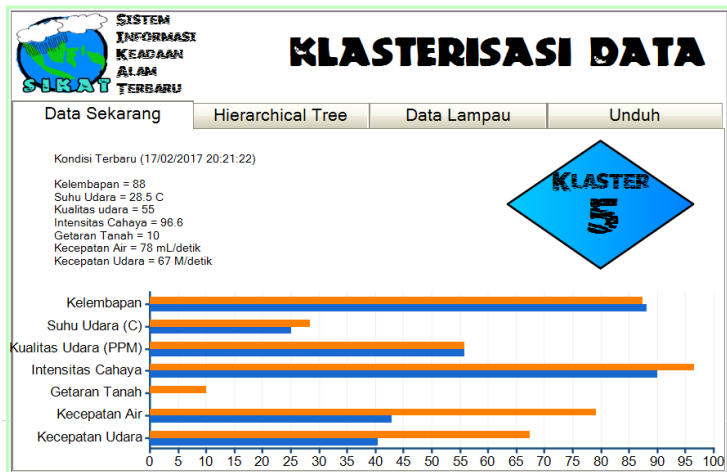
No.	Kebutuhan	Fitur Desain Tampilan
1.	Website dapat menampilkan nilai indikator kondisi keadaan cuaca saat ini.	Adanya tampilan yang menampilkan nilai hasil pengukuran kondisi saat ini.

2	Website dapat menampilkan kluster terdekat dengan data kondisi saat ini.	Adanya tampilan yang menunjukkan kluster terdekat dengan kondisi saat ini, adanya pembeda antara kluster satu dengan kluster lainnya.
3	Website dapat menampilkan perbandingan antara data saat ini dan klaster terdekat.	Adanya sebuah bagan balok yang dapat digunakan sebagai visualisasi perbedaan kondisi saat ini dan kluster terdekat.
4	Website dapat menampilkan Hierarchical Tree dari yang telah dibuat.	Adanya gambar bagan pohon yang mewakili hasil klasterisasi berserta nilai indikator yang digunakan untuk melakukan klasterisasi. Selain itu ada pilihan <i>dropdown</i> yang dapat digunakan untuk memilih hasil klasterisasi pada periode mana yang ingin ditampilkan.
5	Website dapat menampilkan hasil rekapitulasi klasterisasi indikator kondisi keadaan cuaca.	Adanya sebuah table yang dapat menampilkan hasil rekapitulasi klasterisasi data yang pernah dilakukan. Setidaknya table yang dibuat dapat menampilkan

		mayoritas klaster pada setiap klasterisasi.
6	Website dapat memberikan pengguna data hasil pengukuran indikator kondisi cuaca.	Adanya tombol unduh yang dapat digunakan untuk mengunduh data hasil pengukuran. Adanya pilihan seperti kalender agar pengguna dapat menentukan data pada rentang waktu kapan yang akan ia unduh.

4.4.1. Tampilan Data Sekarang

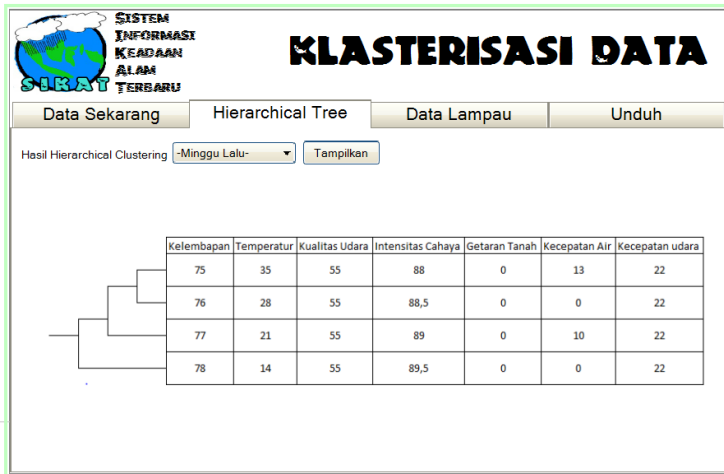
Tampilan ini bertujuan untuk menampilkan kondisi pengukuran indikator cuaca dan perkiraan klasternya jika dibandingkan dengan data hasil klasterisasi minggu lalu. Tampilan Data Sekarang menjadi tampilan utama dari website ini. Tampilan ini memuat poin 1-3 pada Tabel 4.1 Fitur Desain Tampilan Antarmuka. Desain untuk tampilan ini terlihat seperti Gambar 4.1 Desain Tampilan 1.



Gambar 4.1 Desain Tampilan 1

4.4.2. Tampilan Hierarchical Tree


Tampilan ini bertujuan untuk menampilkan secara keseluruhan hasil klasterisasi yang pernah dibuat. Tampilan ini mencakup poin 4 pada Tabel 4.1 Fitur Desain Tampilan Antarmuka. Desain tampilan hierarchical tree terlihat seperti Gambar 4.2 Desain Tampilan 2.



Gambar 4.2 Desain Tampilan 2

4.4.3. Tampilan Data Lampau

Tampilan ini bertujuan untuk memberi informasi kepada pengguna Kluster mana yang menjadi mayoritas dan berapa jumlahnya pada setiap kali dilakukan klasterisasi data setiap minggunya. Tampilan ini memuat poin 5 pada Tabel 4.1 Fitur Desain Tampilan Antarmuka. Bentuk desain tampilan data lampau terlihat seperti Gambar 4.3 Desain Tampilan 3.



**SISTEM
INFORMASI
KEADAAN
ALAM
TERBARU**

KLASTERISASI DATA

Data Sekarang
Hierarchical Tree
Data Lampau
Unduh

	Periode		Kluster	Jumlah
Maret	27/02/2017	05/03/2017	5	350
	06/03/2017	12/03/2017	2	472
	13/03/2017	19/03/2017	3	456
	20/03/2017	26/03/2017	2	251
	27/03/2017	02/04/2017	2	357
April	03/04/2017	09/04/2017	1	456
	10/04/2017	16/04/2017	2	753
	17/04/2017	23/04/2017	2	800
	24/04/2017	30/04/2017	3	158
	01/05/2017	07/05/2017	2	346
Mei	08/05/2017	14/05/2017	2	421
	15/05/2017	21/05/2017	3	315
	22/05/2017	28/05/2017	3	473

Gambar 4.3 Desain Tampilan 3

4.4.4. Tampilan Unduh

Tampilan ini ditujukan kepada pengguna yang ingin mengunduh data hasil pengukuran indikator kondisi cuaca. Pada halaman ini pengguna diberi pilihan untuk dapat menentukan mengunduh data hasil pengukuran pada rentang waktu tertentu. Tampilan ini mencakup poin 6 pada Tabel 4.1 Fitur Desain Tampilan Antarmuka. Desain tampilan unduh terlihat seperti Gambar 4.4 Desain Tampilan 4.

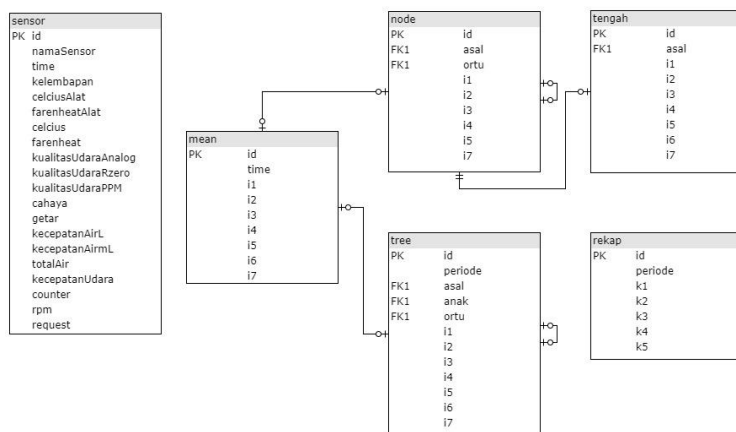


Gambar 4.4 Desain Tampilan 4

4.5. Desain Database

Berdasarkan kebutuhan dari sistem website ini maka dilakukan perancangan database yang akan digunakan. Terdapat 6 tabel yang digunakan oleh website ini yaitu sensor, node, mean, tree, tengah dan rekap seperti yang terlihat pada Gambar 4.5 Desain Database. Tabel sensor berguna untuk menampung semua data hasil pengukuran dari Arduino. Data pada Tabel sensor akan bertambah setiap 10 detik sekali. Tabel mean berfungsi sebagai tempat menyimpan data nilai rata-rata yang dibuat setiap 10 menit sekali yang didapatkan dari data 10 menit terbaru yang ada pada tabel sensor. Tabel 'node' berfungsi sebagai tempat menyimpan data pada proses klasterisasi dan sekaligus menyimpan data klasterisasi terbaru yang dibuat setiap 1 minggu sekali. Data Tabel 'node' akan berubah setiap seminggu sekali yang dimana data yang lama akan dihapus dan diganti dengan data yang baru hasil proses klasterisasi. Tabel tree merupakan tabel yang berfungsi untuk menyimpan data klasterisasi yang telah dilakukan pada waktu lampau. Data pada Tabel tree akan bertambah setiap seminggu sekali sebelum proses klasterisasi yang dimana berasal dari pindahan data pada Tabel 'node'. Pada saat melakukan proses klasterisasi setiap

minggunya maka akan dicari 5 titik tengah yang akan mewakili setiap kluster, 5 titik tengah tersebut akan disimpan pada Tabel tengah. Data pada Tabel tengah inilah yang akan digunakan sebagai perbandingan antara data terbaru untuk dicari sekiranya data tersebut masuk ke kluster mana. Data pada tabel ini akan diperbaharui setiap minggu bersamaan dengan proses klusterisasi. Tabel rekap berfungsi sebagai tempat untuk menyimpan berapa banyak data yang masuk ke dalam kluster ter tentu setiap minggunya.



Gambar 4.5 Desain Database

4.6. Test Case

Untuk melakukan pengujian pada prototipe ketiga maka akan dibuat *test case*. *Test case* sendiri berdasarkan desain tampilan antarmuka, desain database dan kebutuhan fungsional.

Tabel 4.2 Test Case

No.	Sekenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan
1	Data sekarang menampilkan data terbaru pada tabel sensor di database.	Memasukan data baru kedalam tabel sensor pada data base.	Data sekarang menampilkan data yang baru dimasukkan.

2	Data pada tabel mean bertambah setiap sepuluh menit berdasarkan rata-rata data baru pada tabel sensor dalam sepuluh menit terakhir.	Memasukkan 2 data baru kedalam tabel sensor dan tunggu 10 menit.	Ada data baru yang masuk ke tabel mean yang memiliki nilai sama dengan rata-rata dua data baru yang baru saja dimasukkan.
3	Sistem melakukan klasterisasi data terhadap data baru yang masuk dalam seminggu pada tabel mean	Menjalankan event yang memasukkan data dummy setiap sepuluh detik pada tabel sensor. Kemudian menunggu selama 8 hari.	Ada 60.480 data baru pada tabel sensor. Ada 1.008 data baru dalam tabel mean. Terjadi perubahan isi data pada tabel tengah dan node. Ada data baru yang memiliki nilai sama dengan isi data node yang lama pada tabel tree. Ada satu data baru pada tabel rekap.
4	Warna indikator klaster berubah berdasarkan klaster terdekat	Memasukkan data yang memiliki nilai sama dengan kelima titik tengah	Warna indikator klaster berubah setiap data baru masuk.

	dengan data sekarang.	setiap kluster satu persatu.	
5	Bagan balok berubah berdasarkan data terbaru dan kluster terdekat.	Memasukan data baru kedalam tabel sensor.	Bagan balok berubah sesuai dengan data yang dimasukan.
6	Bagan pohon menampilkan hasil klasterisasi terbaru.	Membandingkan bagan pohon yang ditampilkan pada pilihan <i>dropbox</i> “minggu lalu” dengan isi tabel ‘node’.	Isi bagan pohon dan tabel ‘node’ sesuai.
7	Bagan pohon menampilkan hasil klasterisasi yang pernah dilakukan.	Membandingkan isi bagan pohon dengan isi tabel ‘node’ dan tree.	Isi bagan pohon sesuai dengan tabel ‘node’ dan tree.
8	Tabel rekapitulasi data menampilkan data rekapitulasi dengan benar.	Membandingkan isi tabel rekapitulasi data dengan data bapda tabel rekap pada database.	Isi tabel rekapitulasi data sesuai dengan isi tabel rekap di database.
9	Bisa mengunduh isi tabel sensor pada rentang waktu tertentu.	Memilih rentang waktu 01/01/2016 – 01/01/2018 data yang akan diunduh dan klik tombol unduh.	<i>Web browser</i> mengunduh <i>file</i> DataSensor.csv dan memiliki isi sama persis dengan seluruh isi pada tabel sensor.

(Halaman ini sengaja dikosongkan)

BAB V IMPLEMENTASI

Pada bab ini menjelaskan proses implementasi dan pembuatan website ini.

5.1. Lingkungan Implementasi

Lingkungan implementasi dibagi menjadi 2 yaitu lingkungan perangkat keras dan perangkat lunak.

5.1.1. Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan pada pengembangan website ini memiliki *processor* Intel(R) Core(TM) i3-2330M CPU @2.20 GHz (4CPUs) dan *memory* 6144MB RAM.

5.1.2. Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada pengembangan website ini adalah Xampp 7.0.13 sebagai *web service*, Google Chrome 59.0.3071.115 sebagai *web browser* yang digunakan pada saat pengembangan dan pengujian. Sistem operasi yang digunakan pada saat pengembangan maupun pengujian adalah Windows 10 Version 1703 (x64 *based-system*).

5.2. Kode Perangkat Lunak

Kode pada perangkat lunak dikelompokkan menjadi 2 yaitu kode pada database dan kode pada website.

5.2.1. Kode Pada Database

Kode pada database memiliki tujuan utama untuk melakukan proses klasterisasi secara rutin. Kode pada database adalah kode *procedure* dan *event* pada database yang dalam pengembangan perangkat lunak ini menggunakan database MySQL.

5.2.1.1. Kode Untuk Mencari Rata-Rata

Kode 5.1 Kode MySQL CariMean menunjukkan kode pada *event* CariMean. Tujuan dari kode ini adalah menghitung rata-rata dari data yang masuk pada tabel sensor dalam 10 menit terakhir dan memasukkan rata rata tersebut ke dalam tabel mean.

Kode 5.1 Kode MySQL CariMean

```
SET a1 = AVG(kelembapan)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
SET a2 = AVG(cecius)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
SET a3 = AVG(kualitasUdaraPPM)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
SET a4 = AVG(cahaya)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
SET a5 = AVG(getar)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
SET a6 = AVG(kecepatanAirmL)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
SET a7 = AVG(kecepatanUdara)
WHERE time >= (NOW() - INTERVAL 10
MINUTE);
INSERT INTO mean (i1,i2,i3,i4,i5,i6,i7)
VALUES(a1,a2,a3,a4,a5,a6,a7);
```

5.2.1.2. Kode Untuk Melakukan Klasterisasi

Kode ini merupakan kode *procedure* yang memiliki fungsi utama melakukan klasterisasi data pada tabel mean. Parameter yang digunakan pada *procedure* buatNode bernama ‘banyak’ dan memiliki tipe *integer* Cara kerja dari kode ini adalah dengan

terlebih dahulu memindahkan isi tabel 'node' yang merupakan hasil klasterisasi lama ke tabel rekap kemudian mengkosongkan tabel 'node', seperti yang terlihat pada Kode 5.2 Kode MySQL Klasterisasi 1.

Kode 5.2 Kode MySQL Klasterisasi 1

```
INSERT INTO
tree(periode,anak,asal,ortu,i1,i2,i3,i4,i5,i6,i7)
SELECT (NOW() - INTERVAL 1 WEEK), id,
asal, ortu, i1, i2, i3, i4, i5, i6, i7 FROM node;
TRUNCATE node;
```

Setelah mengkosongkan tabel 'node' kemudian menyalin data terbaru sejumlah 'banyak' pada tabel mean ke dalam tabel 'node' berdasarkan pada data terbaru yang ada pada tabel mean seperti yang terlihat pada Kode 5.3 Kode MySQL Klasterisasi 2.

Kode 5.3 Kode MySQL Klasterisasi 2

```
INSERT INTO node (asal,i1,i2,i3,i4,i5,i6,i7)
SELECT id,i1,i2,i3,i4,i5,i6,i7
FROM mean ORDER BY time DESC LIMIT
banyak;
```

Kode 5.4 Kode MySQL Klasterisasi 3 bertujuan melakukan klasterisasi dengan menggunakan pendekatan *Agglomerative* yang dimana perhitungan jarak dilakukan dengan menggunakan rumus *Squared Euclidean distance*. Setelah didapat 5 klaster maka tabel tengah akan dikosongkan dan diganti dengan 5 titik tengah yang ada pada tabel 'node' dan melanjutkan kembali perhitungan sampai hanya terdapat 1 klaster saja.

Kode 5.4 Kode MySQL Klasterisasi 3

```
SET ot = banyak data yang tidak memiliki orang
tua;
WHILE ot > 1
SET pjg = jumlah baris pada tabel 'node';
SET minjrk = -1;
FOR nx = 0; nx < (pjg-1); nx++
SET otx = nilai kolom 'ortu' pada tabel 'node'
baris ke nx;
```

```

IF otx < 1
  SET inx = nilai kolom 'id' pada tabel 'node'
  baris ke nx+1;
  SET a1 = nilai kolom 'i1' pada tabel 'node'
  baris ke nx+1;
  SET a2 = nilai kolom 'i2' pada tabel 'node'
  baris ke nx+1;
  SET a3 = nilai kolom 'i3' pada tabel 'node'
  baris ke nx+1;
  SET a4 = nilai kolom 'i4' pada tabel 'node'
  baris ke nx+1;
  SET a5 = nilai kolom 'i5' pada tabel 'node'
  baris ke nx+1;
  SET a6 = nilai kolom 'i6' pada tabel 'node'
  baris ke nx+1;
  SET a7 = nilai kolom 'i7' pada tabel 'node'
  baris ke nx+1;
FOR ny = nx+1; ny < pjg; ny++
  SET oty = nilai kolom 'ortu' pada tabel
  'node' baris ke ny;
  IF oty < 1
    SET iny = nilai kolom 'id' pada tabel
    'node' baris ke ny+1;
    SET b1 = nilai kolom 'i1' pada tabel
    'node' baris ke ny+1;
    SET b2 = nilai kolom 'i2' pada tabel
    'node' baris ke ny+1;
    SET b3 = nilai kolom 'i3' pada tabel
    'node' baris ke ny+1;
    SET b4 = nilai kolom 'i4' pada tabel
    'node' baris ke ny+1;
    SET b5 = nilai kolom 'i5' pada tabel
    'node' baris ke ny+1;
    SET b6 = nilai kolom 'i6' pada tabel
    'node' baris ke ny+1;
    SET b7 = nilai kolom 'i7' pada tabel
    'node' baris ke ny+1;

```

```

SET jrk = POW((a1-b1),2)+POW((a2-
b2),2)+POW((a3-b3),2)+POW((a4-
b4),2)+POW((a5-b5),2)+POW((a6-
b6),2)+POW((a7-b7),2);
IF (minjrk = -1) OR (jrk < minjrk)
    SET minjrk = jrk;
    SET c1 = (a1+b1)/2;
    SET c2 = (a2+b2)/2;
    SET c3 = (a3+b3)/2;
    SET c4 = (a4+b4)/2;
    SET c5 = (a5+b5)/2;
    SET c6 = (a6+b6)/2;
    SET c7 = (a7+b7)/2;
    SET x = inx;
    SET y = iny;
END IF;
IF (minjrk=0) THEN
    Akhiri perulangan;
END IF;
END IF;
END FOR;
END IF;
END FOR;
INSERT INTO node(asal,ortu,i1,i2,i3,i4,i5,i6,i7)
VALUES (0,0,c1,c2,c3,c4,c5,c6,c7);
SET inn = id data yang baru dimasukan;
UPDATE node SET ortu=inn WHERE id=x;
UPDATE node SET ortu=inn WHERE id=y;
SET ot = banyak data yang tidak memiliki orang
tua;
IF o t= 5 THEN
    TRUNCATE tengah;
    Memasukan data yang tidak memiliki orang tua
    ke dalam tabel 'tengah';
END IF;
END WHILE;

```

Setelah melakukan klasterisasi maka proses selanjutnya adalah menghitung berapa jumlah data pada setiap klaster dan memasukannya pada tabel rekap seperti yang diperlihatkan pada Kode 5.5 Kode MySQL Klasterisasi 4. Proses untuk mencari jumlah anggota tiap klaster dilakukan dengan membandingkan jarak antara data tersebut dan titik tengah klaster sehingga suatu data akan masuk ke dalam klaster yang memiliki titik tengah terdekat.

Kode 5.5 Kode MySQL Klasterisasi 4

```

SET jk1 = 0;
SET jk2 = 0;
SET jk3 = 0;
SET jk4 = 0;
SET jk5 = 0;
FOR ir=1; ir <= banyak; ir++
    SET jk0 = 1;
    SET minjrk = -1;
    SET mk=0;
    SET a1 = nilai kolom 'i1' pada tabel 'node' baris
    ke ir;
    SET a2 = nilai kolom 'i2' pada tabel 'node' baris
    ke ir;
    SET a3 = nilai kolom 'i3' pada tabel 'node' baris
    ke ir;
    SET a4 = nilai kolom 'i4' pada tabel 'node' baris
    ke ir;
    SET a5 = nilai kolom 'i5' pada tabel 'node' baris
    ke ir;
    SET a6 = nilai kolom 'i6' pada tabel 'node' baris
    ke ir;
    SET a7 = nilai kolom 'i7' pada tabel 'node' baris
    ke ir;
    FOR jk0=1; jk0<6; jk0++
        SET b1 = nilai kolom 'i1' pada tabel 'tengah'
        baris ke jk0;

```

```

SET b2 = nilai kolom 'i2' pada tabel 'tengah'
baris ke jk0;
SET b3 = nilai kolom 'i3' pada tabel 'tengah'
baris ke jk0;
SET b4 = nilai kolom 'i4' pada tabel 'tengah'
baris ke jk0;
SET b5 = nilai kolom 'i5' pada tabel 'tengah'
baris ke jk0;
SET b6 = nilai kolom 'i6' pada tabel 'tengah'
baris ke jk0;
SET b7 = nilai kolom 'i7' pada tabel 'tengah'
baris ke jk0;
SET jrk = POW((a1-b1),2)+POW((a2-
b2),2)+POW((a3-b3),2)+POW((a4-
b4),2)+POW((a5-b5),2)+POW((a6-
b6),2)+POW((a7-b7),2);
IF (minjrk=-1 OR jrk<minjrk) THEN
    SET minjrk=jrk;
    SET mk=jk0;
END IF;
END FOR;
IF (mk=1) THEN
    SET jk1=jk1+1;
END IF;
IF (mk=2) THEN
    SET jk2=jk2+1;
END IF;
IF (mk=3) THEN
    SET jk3=jk3+1;
END IF;
IF (mk=4) THEN
    SET jk4=jk4+1;
END IF;
IF (mk=5) THEN
    SET jk5=jk5+1;
END IF;
END FOR;

```

```
INSERT INTO rekap (k1,k2,k3,k4,k5)
VALUES(jk1,jk2,jk3,jk4,jk5);
```

5.2.1.3. Kode Untuk Memulai Klasterisasi

Kode 5.6 Kode MySQL Memulai Klasterisasi merupakan kode *event* yang berjalan seminggu sekali. Tujuan utama dari kode ini adalah untuk memanggil *procedure* buatNode untuk melakukan klasterisasi. Hal pertama yang dilakukan adalah menghitung berapa banyak data baru yang masuk dalam seminggu terakhir pada tabel mean kemudian menjadikan jumlah tersebut parameter untuk memanggil *procedure* buatNode yang merupakan *procedure* untuk melakukan klasterisasi.

Kode 5.6 Kode MySQL Memulai Klasterisasi

```
SET x = banyak data masuk dalam 1 minggu;
CALL buatNode(x);
```

5.2.2. Kode Pada Website

Kode model merupakan kode-kode yang terdapat pada *class* yang menjadi perantara antara *controller* dan database. Ada 5 *class* yang termasuk dalam golongan ini yaitu mNode, mRekap, mSensor, mTengah, dan mTree.

5.2.2.1.1. Kode php Untuk Mengakses Tabel 'node'

Class mNode berguna untuk mengambil seluruh data pada tabel node melalui fungsi getData() yang kodenya seperti terlihat pada *Source Code* PHP Untuk Mengakses Tabel 'node'.

5.2.2.1.2. Kode php Untuk Mengakses Tabel ‘rekap’

Class mRekap berguna untuk mengambil seluruh data pada tabel rekap melalui fungsi *getData()* yang kodenya seperti terlihat pada *Source Code* PHP Untuk Mengakses Tabel ‘rekap’**Error! Reference source not found.**

5.2.2.1.3. Kode php Untuk Mengakses Tabel ‘sensor’

Class mSensor berguna untuk mengambil data terbaru pada tabel sensor melalui fungsi *getNewData()* yang kodenya seperti terlihat pada *Source Code* PHP Untuk Mengakses Tabel ‘sensor’.

5.2.2.1.4. Kode php Untuk Mengakses Tabel ‘tengah’

Class mTree berguna untuk mengambil seluruh data pada tabel tengah melalui fungsi *getData()* yang kodenya seperti terlihat pada *Source Code* PHP Untuk Mengakses Tabel ‘tengah’.

5.2.2.1.5. Kode php Untuk Mengakses Tabel ‘tree’

Class mTree berguna untuk mengambil seluruh jenis periode yang ada pada tabel tree melalui fungsi *getDaftarPeriode()* dan data pada periode tertentu melalui fungsi *getPeriode(\$period)* yang memiliki parameter ‘\$period’ yang berfungsi untuk menentukan data pada periode manayang akan diambil, yang kodenya seperti terlihat pada *Source Code* PHP Untuk Mengakses Tabel ‘tree’.

5.2.2.2. Kode Controller

Kode controller merupakan kode-kode yang terdapat pada *class* yang menjadi pengolah data yang didapat oleh *model* maupun *view* dan mengirimkan hasil olahan data tersebut untuk ditampilkan oleh *view*. Ada 2 *class* yang masuk ke dalam kategori ini yaitu Main dan Unduh.

5.2.2.2.1. Kode php Inti

Class Main merupakan *class* yang paling inti karena pada saat membuka website pertama kali *class* Main lah yang bertugas untuk mengambil data dan mengolahnya untuk menjadi tampilan awal. Setelah data diolah pada *class* Main data tersebut akan dialihkan ke *class* viewTree.

5.2.2.2.2. Kode php Unduh

Class Unduh adalah *class* yang berfungsi untuk mengunduh data pada tabel sensor dalam rentang waktu tertentu, *class* ini dipanggil oleh *class* viewTree. Kode ini bekerja dengan cara menerima masukan dari *class* viewTree yang berisi tanggal awal dan akhir dari data yang akan diunduh kemudian mengirimkan *query* ke *database* untuk mengambil data berdasarkan tanggal masukan tadi, setelah itu melakukan proses pengunduhan *file* yang bernama 'DataSensor.csv' yang berisi hasil dari *database* pada proses sebelumnya.

5.2.2.3. Kode View

Kode view berisi kode-kode dari *class* yang bertugas untuk menampilkan data hasil olahan *controller* sekaligus menerima input dari

pengguna. Terdapat 3 *class* view yaitu *vBar1*, *viewTree* dan *vTree*.

5.2.2.3.1. Kode JavaScript Menampilkan Bagan Balok

File ini merupakan file yang akan ditampilkan pada *viewTree* yang berfungsi untuk menampilkan perbandingan antara data sekarang dan data klaster terdekat. Kode pada *file* ini yang bertujuan untuk membuat tampilan bagan balok dapat dilihat pada *Source Code* PHP, HTML, dan Java Script Untuk Membuat Bagan Balok.

5.2.2.3.2. Kode JavaScript Menampilkan Data

Selain menggunakan bahasa php pada *class* ini juga menggunakan bahasa html dan javascript kode dari *class* *viewTree* yang bertujuan untuk memanggil beberapa file javascript yang dibutuhkan dapat dilihat pada Kode 5.7 Kode JavaScript menampilkan data.

Kode 5.7 Kode JavaScript menampilkan data

```
<script
src="/assets/semantic.js"></script>
<script
src="/assets/semantic.min.js"></script>
<script                src="/assets/jquery-
2.2.4.min.js"></script>
<link rel="stylesheet" type="text/css"
href="/assets/semantic.css">
<link rel="stylesheet" type="text/css"
href="/assets/semantic.min.css">
```

Pada *class* ini memanggil beberapa *class* javascript lain untuk memperindah tampilan

website. Selain itu *class* ini juga mengikut sertakan kode yang terdapat pada *file* vTree.

5.2.2.3.3. Kode JavaScript Menggambar Hirarki

Tujuan dari kode pada *file* ini adalah untuk menampilkan bentuk visualisasi *hierarchical clustering* yang sudah dibuat. Kode yang terdapat pada *file* vTree yang berfungsi untuk membuat tampilan bagan pohon. Sebelum melakukan penampilan data, data yang diambil dari database diubah menjadi *variable array* yang kemudian diubah lagi menjadi *string* yang memiliki format *json* agar lebih mudah unuk menelusuri hubungan antar data dan membuat diagram pohon.

BAB VI PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dibahas proses pengujian dan pembahasan berdasarkan hasil pengujian dari website ini. Pembahasan dan pengujian akan dikelompokkan berdasarkan prototipe yang telah dibuat..

6.1. Pengujian Kecepatan Klasterisasi

Pada pengujian ini dilakukan 6 kali klasterisasi dengan jumlah yang berbeda-beda. Hasil pengukuran dapat dilihat pada Tabel 6.1 Perbandingan Kecepatan Klasterisasi.

Tabel 6.1 Perbandingan Kecepatan Klasterisasi

Jumlah Data	Lama Proses Klasterisasi (detik)
100	130
150	794
200	1.491
250	3.403
360	13.118
1008	78.486

6.2. Pengujian Keakuratan Klasterisasi

Pada pengujian ini dilakukan perbandingan hasil klasterisasi secara manual dan dengan menggunakan sistem. Jumlah data yang digunakan untuk melakukan pengujian berjumlah 5. Data yang digunakan untuk melakukan pengujian berupa data *dummy* yang berisi seperti pada Tabel 6.2 Data Dummy.

Tabel 6.2 Data Dummy

Data	i1	i2	i3	i4	i5	i6	i7
A	50	29	56,23	80	540	541	54
B	51	28	56,23	81	540	541	50
C	49	27	56,23	82	40	541	54

D	53	26	56,23	81	40	541	50
E	47	25	56,23	80	40	541	54

Kemudian dicoba dilakukan klasterisasi dari data *dummy* tersebut sehingga tabel 'node' berisi seperti pada Gambar 6.1 Hasil Klasterisasi. Dimana data A diwakili oleh data yang memiliki id 5, data B diwakili oleh data yang memiliki id 4, data C diwakili oleh data yang memiliki id 3, data D diwakili oleh data yang memiliki id 2 dan data E diwakili oleh data yang memiliki id 1 (baris dalam lingkaran merah). Sedangkan data yang memiliki id 8-11 merupakan node penghubung antar data.

id	asal	ortu	i1	i2	i3	i4	i5	i6	i7
1	1013	8	47	25	56	80	40	541	54
2	1012	10	53	26	56	81	40	541	50
3	1011	8	49	27	56	82	40	541	54
4	1010	9	51	28	56	81	540	541	50
5	1009	9	50	29	56	80	540	541	54
8	0	10	48	26	56	81	40	541	54
9	0	11	50.5	28.5	56	80.5	540	541	52
10	0	11	50.5	26	56	81	40	541	52
11	0	0	50.5	27.25	56	80.75	290	541	52

Gambar 6.1 Hasil Klasterisasi

Kemudian jika dilakukan perhitungan manual untuk mencari data dengan jarak terdekat menggunakan rumus *Sequare Euclidean Distance* maka data C(3) dan data E(1) merupakan data yang paling dekat seperti yang ditampilkan pada Tabel 6.3 Perhitungan Jarak 1.

Tabel 6.3 Perhitungan Jarak 1

	A	B	C	D	E
A	0				

B	19	0			
C	250009	250022	0		
D	250035	250008	34	0	
E	250025	250042	12	54	0

Kemudian data C(3) dan Data E(1) dihubungkan menjadi data dengan nilai tengah seperti pada tabel Tabel 6.4 Node 1. Jika pada sistem maka diwakili oleh data yang memiliki id 8 yang dimana pada tabel ‘node’ data E(1) dan data C(3) memiliki orang tua data yang ber id 8.

Tabel 6.4 Node 1

Data	i1	i2	i3	i4	i5	i6	i7
C-E	48	26	56,23	81	40	541	54

Kemudian dicari lagi data yang memiliki jarak paling dekat. Pada Tabel 6.5 Perhitungan Jarak 2 dapat diketahui bahwa data A(5) dan data B(4) memiliki jarak terdekat sehingga data A(5) dan B(4) dihubungkan menjadi satu data yang memiliki titik tengah seperti pada Tabel 6.6 Node 2, jika pada tabel ‘node’ diwakili oleh data yang memiliki id 9.

Tabel 6.5 Perhitungan Jarak 2

	A	B	C-E	D
A	0			
B	19	0		
C-E	250014	250029	0	
D	250035	250008	41	0

Tabel 6.6 Node 2

Data	i1	i2	i3	i4	i5	i6	i7
A-B	50,5	28,5	56,23	80,5	540	541	52

Kemudian dicari lagi data dengan jarak terdekat, seperti yang terlihat pada Tabel 6.7 Perhitungan Jarak 3 dapat diketahui bahwa data C-E(8) dan data D(2) adalah data yang paling dekat oleh karena itu kedua data tersebut dihubungkan oleh data yang

memiliki titik tengah seperti pada Tabel 6.8 Node 3, jika pada tabel ‘node’ diwakili oleh data yang memiliki id 10.

Tabel 6.7 Perhitungan Jarak 3

	A-B	C-E	D
A-B	0		
C-E	250016,75	0	
D	250016,75	41	0

Tabel 6.8 Node 3

Data	i1	i2	i3	i4	i5	i6	i7
(C-E)-D	50,5	26	56,23	81	40	541	52

Karena hanya tersisa 2 data yang belum terhubung yaitu data A-B(9) dan data C-E-D(10) maka otomatis kedua data tersebut adalah data terdekat. Titik tengah yang menghubungkan kedua data tersebut memiliki nilai seperti pada Tabel 6.9 Node 4, yang jika pada tabel ‘node’ memiliki id 11 yang jika pada Gambar 6.1 Hasil Klasterisasi adalah baris yang dalam lingkaran berwarna hijau.

Tabel 6.9 Node 4

Data	i1	i2	i3	i4	i5	i6	i7
(A-B)- ((C-E)-D)	50,5	27,25	56,23	80,75	290	541	52

6.3. Pengujian Fungsi

Dalam pengujian ini akan dibandingkan antara kebutuhan fungsional dan kondisi sistem sekarang dengan menggunakan *blackbox testing*. Hasil dari perbandingan prototipe ketiga dan *test case* dapat dilihat pada Tabel 6.10 Keluaran Prototipe Ketiga.

Tabel 6.10 Keluaran Prototipe Ketiga

No.	Skenario Pengujian	Keluaran Prototipe Ketiga
1	Data sekarang menampilkan data terbaru pada tabel sensor di database.	Berhasil

2	Data pada tabel mean bertambah setiap sepuluh menit berdasarkan rata-rata data baru pada tabel sensor dalam sepuluh menit terakhir.	Berhasil
3	Sistem melakukan klasterisasi data terhadap data baru yang masuk dalam seminggu pada tabel mean	Berhasil
4	Warna indikator klaster berubah berdasar kan klaster terdekat dengan data sekarang.	Berhasil
5	Bagan balok berubah berdasarkan data terbaru dan klaster terdekat.	Berhasil
6	Bagan pohon menampilkan hasil klasterisasi terbaru.	Berhasil
7	Bagan pohon menampilkan hasil klasterisasi yang pernah dilakukan.	Berhasil
8	Tabel rekapitulasi data menampilkan data rekapitulasi dengan benar.	Berhasil
9	Bisa ngunduh isi tabel sensor pada rentang waktu tertentu.	Berhasil

6.4. Pembahasan

Mengacu pada hasil pengujian kecepatan klasterisasi prototipe pertama bahwa sistem tidak layak untuk melakukan klasterisasi data dalam jumlah banyak dan dalam waktu yang singkat. Akan tetapi pada sistem ini proses klasterisasi dilakukan seminggu sekali sehingga ada jarak waktu 1 minggu antara proses klasterisasi itu berarti ada waktu 604.800 detik untuk menyelesaikan proses klasterisasi. Data yang masuk pada sistem memiliki jeda 10 detik dan setiap 10 menit data akan dirata-rata dan proses klasterisasi akan menggunakan data hasil rata-rata yaitu sebanyak 1.008 data, mengacu pada Tabel 6.1

Perbandingan Kecepatan Klasterisasi bahwa untuk memproses 1.008 data memerlukan waktu 78.486 sehingga sangat memungkinkan untuk sistem dapat bertahan melakukan klasterisasi setiap minggunya. Mengacu pada hasil pengujian keakuratan Klasterisasi dapat dilihat bahwa hasil dari klasterisasi sistem sama dengan hasil klasterisasi manual sehingga bisa dikatakan hasil klasterisasi dari sistem akurat. Sedangkan jika dilihat dari pengujian fungsi maka dapat dilihat jika sistem dapat melakukan semua fungsi yang ada karena berhasil melakukan apa saja yang ada pada *test case*.

BAB VII

KESIMPULAN DAN SARAN

Pada bab ini akan menampilkan hasil kesimpulan dan saran dari penelitian ini.

7.1. Kesimpulan

Berdasarkan hasil pengujian maka dapat disimpulkan hal sebagai berikut:

- Sistem tidak dapat melakukan klasterisasi data dalam jumlah yang terlalu banyak dalam waktu yang cepat untuk itu data yang diolah harus dikurangi dan jarak antar proses klasterisasi harus diperbesar.
- Untuk mengurangi jumlah data yang diolah maka data yang masuk akan dirata-rata setiap 10 menit dan hasil rata-rata tersebut yang akan diolah.
- Untuk memperbesar jarak antar proses klasterisasi maka data akan diklasterisasi setiap 1 minggu sekali dan data yang ditampilkan merupakan perbandingan data sekarang dan hasil klasterisasi minggu lalu.
- Sistem dapat melakukan proses klasterisasi dengan menghitung jarak menggunakan *Squared Euclidean distance* dan pendekatan *Agglomerative*.
- Sistem dapat melakukan semua yang ada pada kebutuhan fungsional maka dapat dikatakan sistem sudah sesuai dengan kebutuhan.

7.2. Saran

Saran yang dapat dilakukan untuk pengembangan sistem ini di masa yang akan datang adalah:

- Mencoba menggunakan metode lain untuk melakukan klasterisasi.
- Melakukan klasterisasi dengan menggunakan perangkat keras yang lebih baik.

- Melakukan pembenahan dari segi desain *user interface* yang lebih *user friendly*.

DAFTAR PUSTAKA

- [1] H. M. Jess Christopher B. Lopez, "Low-Cost Weather Monitoring System with Online Logging and Data Visualization," *8th IEEE International Conference Humanoid*, vol. 8, no. 1, pp. 1-6, 20015.
- [2] C. Y. Kilaso, "A Wireless Sensor Network for Weather and Disaster Alarm Systems," *International Conference on Information and Electronics Engineering*, pp. 1-5, 2011.
- [3] Rajendra Kumar Roul, Shubham Rohan Asthana, Sanjay Kumar Sahay, "Automated Document Indexing via Intelligent Hierarchical Clustering: A Novel Approach," *ornell University Library*, pp. 1-6, 2015.
- [4] Guo Jun Gan, Chao Qun Ma, Jian Hong Wu, *Data Clustering: Theory, Algorithms, And Applications*, Beijing: ASA-SIAM, 2007.
- [5] O. A. Abb, "Comparison Between data Clustering Algorithm," *The International Arab Journal of Information Technology*, vol. 5, pp. 320-325, 2008.
- [6] D. S. Rafi'i, *Meteorologi dan Klimatologi*, Jakarta: Angkasa, 1995.
- [7] S. Komatineni, "Reshaping IT Project Delivery Through Extreme Prototyping," November 2006. [Online]. Available: <http://www.onjava.com/pub/a/onjava/2006/11/15/reshaping-it-project-delivery-through-extreme-prototyping.html>.

LAMPIRAN

Source Code MySQL Event Membuat Rata-Rata Setiap 10 Menit

```
CREATE      DEFINER=`root`@`localhost`      EVENT
`CariMean` ON SCHEDULE EVERY 10 MINUTE
STARTS '2017-01-01 00:00:00' ENDS '2027-01-01
00:00:00' ON COMPLETION NOT PRESERVE ENABLE
DO BEGIN
DECLARE a1 double(7,2);
DECLARE a2 double(7,2);
DECLARE a3 double(7,2);
DECLARE a4 double(7,2);
DECLARE a5 double(7,2);
DECLARE a6 double(7,2);
DECLARE a7 double(7,2);
SELECT AVG(kelembapan) FROM sensor WHERE time
>= (NOW() - INTERVAL 10 MINUTE) into a1;
SELECT AVG(cecius) FROM sensor WHERE time >=
(NOW() - INTERVAL 10 MINUTE) into a2;
SELECT AVG(kualitasUdaraPPM) FROM sensor WHERE
time >= (NOW() - INTERVAL 10 MINUTE) into a3;
SELECT AVG(cahaya) FROM sensor WHERE time >=
(NOW() - INTERVAL 10 MINUTE) into a4;
SELECT AVG(getar) FROM sensor WHERE time >=
(NOW() - INTERVAL 10 MINUTE) into a5;
SELECT AVG(kecepatanAirmL) FROM sensor WHERE
time >= (NOW() - INTERVAL 10 MINUTE) into a6;
SELECT AVG(kecepatanUdara) FROM sensor WHERE
time >= (NOW() - INTERVAL 10 MINUTE) into a7;
INSERT INTO mean (i1,i2,i3,i4,i5,i6,i7)
VALUES(a1,a2,a3,a4,a5,a6,a7);
END
```

Source Code MySQL Procedure Klasterisasi

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`buatNode`(IN `banyak` INT)
    NO SQL
BEGIN
DECLARE a1 double(7,2);
DECLARE a2 double(7,2);
DECLARE a3 double(7,2);
DECLARE a4 double(7,2);
DECLARE a5 double(7,2);
DECLARE a6 double(7,2);
DECLARE a7 double(7,2);
DECLARE b1 double(7,2);
DECLARE b2 double(7,2);
DECLARE b3 double(7,2);
DECLARE b4 double(7,2);
DECLARE b5 double(7,2);
DECLARE b6 double(7,2);
DECLARE b7 double(7,2);
DECLARE c1 double(7,2);
DECLARE c2 double(7,2);
DECLARE c3 double(7,2);
DECLARE c4 double(7,2);
DECLARE c5 double(7,2);
DECLARE c6 double(7,2);
DECLARE c7 double(7,2);
DECLARE jrk double(7,2);
DECLARE minjrk double(7,2);
DECLARE pjg int;
DECLARE nx int;
DECLARE ny int;
DECLARE x int;
DECLARE y int;
DECLARE inn int;
DECLARE inx int;
DECLARE iny int;
```

```

DECLARE jr int;
DECLARE ir int;
DECLARE mk int;
DECLARE jk0 int;
DECLARE jk1 int;
DECLARE jk2 int;
DECLARE jk3 int;
DECLARE jk4 int;
DECLARE jk5 int;
INSERT                                                                    INTO
tree(periode,anak,asal,ortu,i1,i2,i3,i4,i5,i6,i7)
SELECT      (NOW()      -      INTERVAL      1
WEEK),id,asal,ortu,i1,i2,i3,i4,i5,i6,i7 FROM node;
TRUNCATE node;
INSERT INTO node (asal,i1,i2,i3,i4,i5,i6,i7)
SELECT id,i1,i2,i3,i4,i5,i6,i7 FROM mean ORDER BY
time DESC LIMIT banyak;
SELECT COUNT(*) from node INTO pjg;
loop0: WHILE (pjg<((banyak*2)-1)) DO
SELECT COUNT(*) from node INTO pjg;
SET minjrk=-1;
set nx=0;
loop1: WHILE (nx<(pjg-1)) DO
SELECT ortu FROM node LIMIT nx,1 INTO jr;
IF (jr<1) THEN
SELECT id FROM node LIMIT nx,1 INTO inx;
SELECT i1 FROM node LIMIT nx,1 INTO a1;
SELECT i2 FROM node LIMIT nx,1 INTO a2;
SELECT i3 FROM node LIMIT nx,1 INTO a3;
SELECT i4 FROM node LIMIT nx,1 INTO a4;
SELECT i5 FROM node LIMIT nx,1 INTO a5;
SELECT i6 FROM node LIMIT nx,1 INTO a6;
SELECT i7 FROM node LIMIT nx,1 INTO a7;
SET ny=nx+1;
loop2: WHILE (ny<pjg) DO
SELECT ortu FROM node LIMIT ny,1 INTO jr;
IF (jr<1) THEN

```



```

SELECT id FROM node LIMIT ny,1 INTO iny;
SELECT i1 FROM node LIMIT ny,1 INTO b1;
SELECT i2 FROM node LIMIT ny,1 INTO b2;
SELECT i3 FROM node LIMIT ny,1 INTO b3;
SELECT i4 FROM node LIMIT ny,1 INTO b4;
SELECT i5 FROM node LIMIT ny,1 INTO b5;
SELECT i6 FROM node LIMIT ny,1 INTO b6;
SELECT i7 FROM node LIMIT ny,1 INTO b7;
SET jrk = POW((a1-b1),2)+POW((a2-b2),2)+POW((a3-
b3),2)+POW((a4-b4),2)+POW((a5-b5),2)+POW((a6-
b6),2)+POW((a7-b7),2);
IF (minjrk=-1 OR jrk<minjrk) THEN
SET minjrk=jrk;
SET c1=(a1+b1)/2;
SET c2=(a2+b2)/2;
SET c3=(a3+b3)/2;
SET c4=(a4+b4)/2;
SET c5=(a5+b5)/2;
SET c6=(a6+b6)/2;
SET c7=(a7+b7)/2;
SET x=inx;
SET y=iny;
END IF;
IF (minjrk=0) THEN
SET ny=pjg-1;
SET nx=pjg-2;
END IF;
END IF;
SET ny=ny+1;
END WHILE loop2;
END IF;
SET nx=nx+1;
END WHILE loop1;
INSERT INTO node(asal,ortu,i1,i2,i3,i4,i5,i6,i7)
VALUES (0,0,c1,c2,c3,c4,c5,c6,c7);
SELECT id FROM node ORDER BY id DESC limit 1 INTO
inn;

```

```

UPDATE node SET ortu=inn WHERE id=x;
UPDATE node SET ortu=inn WHERE id=y;
SELECT COUNT(*) from node INTO pjg;
IF(pjg=5)THEN
TRUNCATE tengah;
INSERT INTO tengah(asal,i1,i2,i3,i4,i5,i6,i7)
SELECT id,i1,i2,i3,i4,i5,i6,i7 FROM node WHERE ortu=0;
END IF;
END WHILE loop0;
SET ir=1;
SET jk0=0;
SET jk1=0;
SET jk2=0;
SET jk3=0;
SET jk4=0;
SET jk5=0;
WHILE (ir<(banyak+1)) DO
SET jk0=1;
SET minjrk=-1;
SET mk=0;
SELECT i1 FROM node WHERE id=ir INTO a1;
SELECT i2 FROM node WHERE id=ir INTO a2;
SELECT i3 FROM node WHERE id=ir INTO a3;
SELECT i4 FROM node WHERE id=ir INTO a4;
SELECT i5 FROM node WHERE id=ir INTO a5;
SELECT i6 FROM node WHERE id=ir INTO a6;
SELECT i7 FROM node WHERE id=ir INTO a7;
WHILE (jk0<6) DO
SELECT i1 FROM tengah WHERE id=jk0 INTO b1;
SELECT i2 FROM tengah WHERE id=jk0 INTO b2;
SELECT i3 FROM tengah WHERE id=jk0 INTO b3;
SELECT i4 FROM tengah WHERE id=jk0 INTO b4;
SELECT i5 FROM tengah WHERE id=jk0 INTO b5;
SELECT i6 FROM tengah WHERE id=jk0 INTO b6;
SELECT i7 FROM tengah WHERE id=jk0 INTO b7;

```

```

SET jrk = POW((a1-b1),2)+POW((a2-b2),2)+POW((a3-
b3),2)+POW((a4-b4),2)+POW((a5-b5),2)+POW((a6-
b6),2)+POW((a7-b7),2);
IF (minjrk=-1 OR jrk<minjrk) THEN
SET minjrk=jrk;
SET mk=jk0;
END IF;
SET jk0=jk0+1;
END WHILE;
IF (mk=1) THEN
SET jk1=jk1+1;
END IF;
IF (mk=2) THEN
SET jk2=jk2+1;
END IF;
IF (mk=3) THEN
SET jk3=jk3+1;
END IF;
IF (mk=4) THEN
SET jk4=jk4+1;
END IF;
IF (mk=5) THEN
SET jk5=jk5+1;
END IF;
SET ir=ir+1;
END WHILE;
INSERT INTO rekap (k1,k2,k3,k4,k5)
VALUES(jk1,jk2,jk3,jk4,jk5);
END

```

Source Code MySQL Event Memulai Proses Klasterisasi Setiap Minggu

```
CREATE      DEFINER=`root`@`localhost`      EVENT
`BuatTree` ON SCHEDULE EVERY 1 WEEK STARTS
'2017-01-01 00:00:00' ENDS '2027-01-01 00:00:00' ON
COMPLETION NOT PRESERVE ENABLE DO BEGIN
DECLARE banyak int;
SELECT COUNT(*) FROM mean WHERE time >= NOW()
- INTERVAL 1 WEEK INTO banyak;
CALL buatNode(banyak);
END
```

Source Code PHP Untuk Mengakses Tabel ‘node’

```
<?php

class mNode extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function getData() {
        $this->db->select('id as a, asal as b, ortu as c, i1, i2, i3,
i4, i5, i6, i7, mean.time as d');
        $this->db->from('node');
        $this->db->join('mean', 'mean.id = node.asal', 'left');
        $this->db->order_by('ortu','ASC');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
            return $row;
        }
    }
}
```

Source Code PHP Untuk Mengakses Tabel ‘rekap’

```
<?php

class mRekap extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function getData() {
        $this->db->select('periode,k1,k2,k3,k4,k5');
        $this->db->from('rekap');
        $this->db->order_by('periode', 'DESC');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
            return $row;
        }
    }
}
```

Source Code PHP Untuk Mengakses Tabel ‘sensor’

```

<?php

class mSensor extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function getData() {
        $this->db->select('id a, time b, kelembapan c, celcius d,
kualitasUdaraPPM e, cahaya f, getar g, kecepatanAirmL h,
kecepatanUdara i');
        $this->db->from('sensor');
        $this->db->limit(1008);
        $this->db->order_by('time ASC');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
        } else {
            return FALSE;
        }
    }

    function getNewData() {
        $this->db->select('id a, time b, kelembapan c, celcius d,
kualitasUdaraPPM e, cahaya f, getar g, kecepatanAirmL h,
kecepatanUdara i');
        $this->db->from('sensor');
        $this->db->limit(1);
        $this->db->order_by('time DESC');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
        }
    }
}

```

```
    } else {  
        return FALSE;  
    }  
}  
}
```


Source Code PHP Untuk Mengakses Tabel ‘tengah’

```

<?php

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of mTengah
 *
 * @author Akaga
 */
class mTengah extends CI_Model{
    function __construct() {
        parent::__construct();
    }

    function getMax(){
        $this->db->select('Max(i1) a, Max(i2) b, Max(i3) c,
Max(i4) d, Max(i5) e, Max(i6) f, Max(i7) g');
        $this->db->from('tengah');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
        } else {
            return FALSE;
        }
    }

    function getMin(){
        $this->db->select('Min(i1) a, Min(i2) b, Min(i3) c,
Min(i4) d, Min(i5) e, Min(i6) f, Min(i7) g');

```

```

$this->db->from('tengah');
$query = $this->db->get();
if ($query->num_rows() > 0) {
    $r = $query->result();
    return $r;
} else {
    return FALSE;
}
}

function get3($in){
    $this->db->select($in);
    $this->db->from('tengah');
    $this->db->order_by($in,'ASC');
    $this->db->limit(1, 2);
    $query = $this->db->get();
    if ($query->num_rows() > 0) {
        $r = $query->result();
        return $r;
    } else {
        return FALSE;
    }
}

function getData(){
    $this->db->select('id a, asal b, i1 c, i2 d, i3 e, i4 f, i5 g,
i6 h, i7 i');
    $this->db->from('tengah');
    $query = $this->db->get();
    if ($query->num_rows() > 0) {
        $r = $query->result();
        return $r;
    } else {
        return FALSE;
    }
}
}

```

}

Source Code PHP Untuk Mengakses Tabel ‘tree’

```

<?php
class mTree extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function getDaftarPeriode() {
        $this->db->distinct();
        $this->db->select('periode as a');
        $this->db->from('tree');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
        } else {
            return FALSE;
        }
    }

    function getPeriode($period) {
        $kondisi = 'periode="' . $period . '"';
        $this->db->select('anak j,asal a,ortu b,i1 c,i2 d,i3 e,i4
f,i5 g,i6 h,i7 i');
        $this->db->from('tree');
        $this->db->where($kondisi);
        $this->db->order_by('ortu','ASC');
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $r = $query->result();
            return $r;
        } else {
            return FALSE;
        }
    }
}

```

Source Code PHP Controller Main

```

<?php

ini_set('max_execution_time', -1);

defined('BASEPATH') OR exit('No direct script access
allowed');

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of Main
 *
 * @author Akaga
 */
class Main extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('mNode', "", TRUE);
        $this->load->model('mSensor', "", TRUE);
        $this->load->model('mTengah', "", TRUE);
        $this->load->model('mTree', "", TRUE);
        $this->load->model('mRekap', "", TRUE);
    }

    public function index() {
        $data = array();
        $dataNode = array();
        $jNode = 0;
        $result = $this->mNode->getData();

```

```

foreach ($result as $row) {
    $temp = array();
    array_push($temp, $row->a);
    array_push($temp, $row->b);
    array_push($temp, $row->c);
    array_push($temp, $row->i1);
    array_push($temp, $row->i2);
    array_push($temp, $row->i3);
    array_push($temp, $row->i4);
    array_push($temp, $row->i5);
    array_push($temp, $row->i6);
    array_push($temp, $row->i7);
    array_push($temp, $row->d);
    array_push($dataNode, $temp);
    $jNode++;
}
$newData = array();
$result = $this->mSensor->getNewData();
foreach ($result as $row) {
    $temp = array();
    array_push($temp, $row->a);
    array_push($temp, $row->b);
    array_push($temp, $row->c);
    array_push($temp, $row->d);
    array_push($temp, $row->e);
    array_push($temp, $row->f);
    array_push($temp, $row->g);
    array_push($temp, $row->h);
    array_push($temp, $row->i);
    array_push($newData, $temp);
}
$dataTengah = array();
$result = $this->mTengah->getData();
foreach ($result as $row) {
    $temp = array();
    array_push($temp, $row->a);
    array_push($temp, $row->b);

```

```

        array_push($temp, $row->c);
        array_push($temp, $row->d);
        array_push($temp, $row->e);
        array_push($temp, $row->f);
        array_push($temp, $row->g);
        array_push($temp, $row->h);
        array_push($temp, $row->i);
        array_push($dataTengah, $temp);
    }
    $min = 0;
    $klaster = 0;
    for ($i = 0; $i < 5; $i++) {
        $jrk = 0;
        for ($j = 2; $j < 9; $j++) {
            $jrk = $jrk + (pow(($newData[0][$j] -
            $dataTengah[$i][$j]), 2));
        }
        if ($i < 1 || $jrk < $min) {
            $min = $jrk;
            $klaster = $i + 1;
        }
    }
    $dPeriode = array();
    $hTree = array();
    $jdPeriode = 0;
    $result = $this->mTree->getDaftarPeriode();
    foreach ($result as $row) {
        array_push($dPeriode, $row->a);
        $jdPeriode++;
    }
    for ($i = 0; $i < $jdPeriode; $i++) {
        $p = $dPeriode[$i];
        $temp0 = array();
        $result = $this->mTree->getPeriode($p);
        foreach ($result as $row) {
            $temp1 = array();
            array_push($temp1, $row->j);
        }
    }

```

```

        array_push($temp1, $row->a);
        array_push($temp1, $row->b);
        array_push($temp1, $row->c);
        array_push($temp1, $row->d);
        array_push($temp1, $row->e);
        array_push($temp1, $row->f);
        array_push($temp1, $row->g);
        array_push($temp1, $row->h);
        array_push($temp1, $row->i);
        array_push($temp1, $row->j);
        array_push($temp0, $temp1);
    }
    array_push($hTree, $temp0);
}
$dataRekap = array();
$jRekap = 0;
$result = $this->mRekap->getData();
foreach ($result as $row) {
    $temp = array();
    array_push($temp, $row->periode);
    array_push($temp, $row->k1);
    array_push($temp, $row->k2);
    array_push($temp, $row->k3);
    array_push($temp, $row->k4);
    array_push($temp, $row->k5);
    array_push($dataRekap, $temp);
    $jRekap++;
}
$minTengah = array();
$result = $this->mTengah->getMin();
foreach ($result as $row) {
    array_push($minTengah, $row->a);
    array_push($minTengah, $row->b);
    array_push($minTengah, $row->c);
    array_push($minTengah, $row->d);
    array_push($minTengah, $row->e);
    array_push($minTengah, $row->f);

```



```

        array_push($minTengah, $row->g);
    }
    $maxTengah = array();
    $result = $this->mTengah->getMax();
    foreach ($result as $row) {
        array_push($maxTengah, $row->a);
        array_push($maxTengah, $row->b);
        array_push($maxTengah, $row->c);
        array_push($maxTengah, $row->d);
        array_push($maxTengah, $row->e);
        array_push($maxTengah, $row->f);
        array_push($maxTengah, $row->g);
    }
    $deskripsi="";
    $result = $this->mTengah->get3('i1');
    foreach ($result as $row){
        $data3=$row->i1;
        $dataT=$dataTengah[$klaster-1][2];
        if($dataT==$data3){
            $deskripsi=$deskripsi.'Kelembapan udara biasa-
biasa saja, ';
        }elseif ($dataT<$data3) {
            if($dataT>$minTengah[0]){
                $deskripsi=$deskripsi.'Kelembapan udara agak
rendah, ';
            } else {
                $deskripsi=$deskripsi.'Kelembapan udara
rendah, ';
            }
        } else {
            if ($dataT<$maxTengah[0]) {
                $deskripsi=$deskripsi.'Kelembapan udara agak
tinggi, ';
            } else {
                $deskripsi=$deskripsi.'Kelembapan udara
tinggi, ';
            }
        }
    }

```

```

    }
}
$result = $this->mTengah->get3('i2');
foreach ($result as $row){
    $data3=$row->i2;
    $dataT=$dataTengah[$klaster-1][3];
    if($dataT==$data3){
        $deskripsi=$deskripsi.'Suhu udara biasa-biasa
saja, ';
    }elseif ($dataT<$data3) {
        if($dataT>$minTengah[1]){
            $deskripsi=$deskripsi.'Suhu udara agak dingin,
';
        } else {
            $deskripsi=$deskripsi.'Suhu udara dingin, ';
        }
    } else {
        if ($dataT<$maxTengah[1]) {
            $deskripsi=$deskripsi.'Suhu udara agak panas, ';
        } else {
            $deskripsi=$deskripsi.'Suhu udara panas, ';
        }
    }
}
}
$result = $this->mTengah->get3('i3');
foreach ($result as $row){
    $data3=$row->i3;
    $dataT=$dataTengah[$klaster-1][4];
    if($dataT==$data3){
        $deskripsi=$deskripsi.'Kualitas udara biasa-biasa
saja, ';
    }elseif ($dataT<$data3) {
        if($dataT>$minTengah[2]){
            $deskripsi=$deskripsi.'Kualitas udara agak
buruk, ';
        } else {
            $deskripsi=$deskripsi.'Kualitas udara buruk, ';

```

```

    }
  } else {
    if ($dataT<$maxTengah[2]) {
      $deskripsi=$deskripsi.'Kualitas udara agak baik,
';
    } else {
      $deskripsi=$deskripsi.'Kualitas udara baik, '
    }
  }
}
$result = $this->mTengah->get3('i4');
foreach ($result as $row){
  $data3=$row->i4;
  $dataT=$dataTengah[$klaster-1][5];
  if($dataT==$data3){
    $deskripsi=$deskripsi.'Langit tidak terlalu
berawan, '
  }elseif ($dataT<$data3) {
    if($dataT>$minTengah[3]){
      $deskripsi=$deskripsi.'Langit agak berawan, '
    } else {
      $deskripsi=$deskripsi.'Langit berawan, '
    }
  } else {
    if ($dataT<$maxTengah[3]) {
      $deskripsi=$deskripsi.'Langit cerah, '
    } else {
      $deskripsi=$deskripsi.'Langit sangat cerah, '
    }
  }
}
$result = $this->mTengah->get3('i5');
foreach ($result as $row){
  $data3=$row->i5;
  $dataT=$dataTengah[$klaster-1][6];
  if($dataT==$minTengah[4]){
    $deskripsi=$deskripsi.'Tanah tidak begetar, '
  }
}

```

```

    }elseif ($dataT<$data3) {
        if($dataT<$data3){
            $deskripsi=$deskripsi.'Langit begetar sedikit, ';
        } else {
            $deskripsi=$deskripsi.'Langit agak begetar, ';
        }
    } else {
        if ($dataT<$maxTengah[4]) {
            $deskripsi=$deskripsi.'Tanah begetar, ';
        } else {
            $deskripsi=$deskripsi.'Tanah sangat begetar, ';
        }
    }
}
$result = $this->mTengah->get3('i6');
foreach ($result as $row){
    $data3=$row->i6;
    $dataT=$dataTengah[$klaster-1][7];
    if($dataT==$data3){
        $deskripsi=$deskripsi.'Air mengalir biasa saja, ';
    }elseif ($dataT<$data3) {
        if($dataT>$minTengah[5]){
            $deskripsi=$deskripsi.'Air mengalir agak
lambat, ';
        } else {
            $deskripsi=$deskripsi.'Air mengalir lambat, ';
        }
    } else {
        if ($dataT<$maxTengah[5]) {
            $deskripsi=$deskripsi.'Air mengalir agak cepat,
';
        } else {
            $deskripsi=$deskripsi.'Air mengalir cepat ';
        }
    }
}
$result = $this->mTengah->get3('i7');

```

```

foreach ($result as $row){
    $data3=$row->i7;
    $dataT=$dataTengah[$klaster-1][8];
    if($dataT==$data3){
        $deskripsi=$deskripsi.'Angin sepoi-sepoi.';
    }elseif ($dataT<$data3) {
        if($dataT>$minTengah[6]){
            $deskripsi=$deskripsi.'Angin          berhembus
pelan.';
        } else {
            $deskripsi=$deskripsi.'Angin berhembus sangat
pelan.';
        }
    } else {
        if ($dataT<$maxTengah[6]) {
            $deskripsi=$deskripsi.'Angin          berhembus
kencang.';
        } else {
            $deskripsi=$deskripsi.'Angin berhembus sangat
kencang.';
        }
    }
}
}
$data['deskripsi']=$deskripsi;
$data['dataRekap'] = $dataRekap;
$data['jRekap'] = $jRekap;
$data['hTree'] = $hTree;
$data['dPeriode'] = $dPeriode;
$data['jdPeriode'] = $jdPeriode;
$data['dataTengah'] = $dataTengah;
$data['klaster'] = $klaster;
$data['newData'] = $newData;
$data['dataNode'] = $dataNode;
$data['jNode'] = $jNode;
$this->load->view('viewTree', $data);
}
}

```


Source Code PHP Controller Unduh

```

<?php
defined('BASEPATH') OR exit('No direct script access
allowed');

/**
 * Description of UnduhData
 *
 * @author Akaga
 */
class unduh extends CI_Controller{
    function __construct() {
        parent::__construct();
    }
    public function index(){
        $date=$this->input->post('awal');
        $awal=date('Y-m-d H:i:s', strtotime($date));
        $date=$this->input->post('akhir');
        $akhir=date('Y-m-d H:i:s', strtotime($date));
        $qwr='SELECT * FROM sensor WHERE
time>"'.$awal.'" AND time<"'.$akhir.'"';
        $this->load->dbutil();
        $this->load->helper('download');
        $query = $this->db->query($qwr);
        $filename = 'DataSensor.csv';
        force_download($filename, $this->dbutil-
>csv_from_result($query));
    }
}

```

Source Code PHP, HTML, dan Java Script Tampilan Inti

```

<!DOCTYPE html>
<!--
To change this license header, choose License Headers in
Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <meta charset="UTF-8">
    <title>SIKAT -Klasterisasi Data- </title>
    <link      rel="icon"      href="/assets/Sikat.jpg"
type="image/x-icon">
    <script src="/assets/semantic.js"></script>
    <script src="/assets/semantic.min.js"></script>
    <script src="/assets/jquery-2.2.4.min.js"></script>
    <link      rel="stylesheet"      type="text/css"
href="/assets/semantic.css">
    <link      rel="stylesheet"      type="text/css"
href="/assets/semantic.min.css">
  </head>
  <body>
    <style type="text/css">
      @font-face {
        font-family: "fontON";
        src: url(/assets/Toucon_.ttf) format("truetype");
      }
      p.customON {
        font-family: "fontON", Verdana, Tahoma;
      }
    </style>

    <div class="item">
      <div><br></div>
      <div class="ui stackable grid">
        <div class="ui small image">

```



```

        
    </div>
    <div class="middle aligned content">
        <h3 class="header">

Sistem<br>Informasi<br>Keadaan<br>Alam<br>Terbaru
        </h3>
    </div>
    <p      class="customON"      style="font-
size:75px">Klasterisasi Data</p>
    </div>
</div>

<style>
    g.node {
        font-family: Verdana, Helvetica;
        font-size: 12px;
        font-weight: bold;
    }

    circle.node-dot {
        fill: blue;
        stroke: red;
        stroke-width: 1px;
    }

    path.link {
        fill: none;
        stroke: gray;
    }
</style>

<div id="DaftarMenu" class="ui three item menu">
    <a id="Tab1" class="item" >Kondisi Terbaru</a>
    <a id="Tab2" class="item" >Hierarchical
Clustering</a>
    <a id="Tab3" class="item" >Rekapitulasi Data</a>

```

```

</div>
<div id="Isi1" class="ui bottom attached tab segment">
  <div class="ui stackable grid">
    <div class="ui ten wide column">
      <?php
        echo '<p class="customON" style="font-size:20px">';
        echo 'Kondisi terbaru (' . $newData[0][1] .
      '<br>';
        echo '</p>';
        echo 'Kelembapan = ' . $newData[0][2] . '<br>';
        echo 'Suhu Udara = ' . $newData[0][3] . 'C<br>';
        echo 'Kualitas Udara = ' . $newData[0][4] .
      '<br>';
        echo 'Intensitas Cahaya = ' . $newData[0][5] .
      '<br>';
        echo 'Getaran Tanah = ' . $newData[0][6] .
      '<br>';
        echo 'Kecepatan Air = ' . $newData[0][7] . '
      mL/detik<br>';
        echo 'Kecepatan Udara = ' . $newData[0][8] . '
      M/detik<br>';
        ?>
      </div>
    <div class="ui center aligned six wide column">
      <p class="customON" style="font-size:25px">Mirip Dengan Kondisi:
      <div class="ui <?php
        if ($klaster == 1) {
          echo 'red';
        } else if ($klaster == 2) {
          echo 'yellow';
        } else if ($klaster == 3) {
          echo 'green';
        } else if ($klaster == 4) {
          echo 'blue';
        } else {

```

```

        echo 'violet';
    }
    ?> circular label">
        <div class="ui stackable grid">
            <div class="ui one wide column"></div>
            <div>
                <p class="customON" style="font-
size:30px">
                    <br>klaster<br>
                    <?php echo $klaster;
?><br></br></p>
                </div>
            <div class="ui one wide column"></div>
        </div>
    </div>
    <p class="customON" style="font-size:15px">
        Deskripsi kondisi klaster <?php echo $klaster;
?>:<br>
        <h5><?php echo $deskripsi; ?></h5>
    </p>

    <div>
        <?php include('vBar1.php ') ?>
    </div>

    <div id="Isi2" class="ui bottom attached tab segment
active">

        <div class="ui stackable grid">
            <?php include('vTree.php ') ?>
        </div>

    </div>

```

```

<div id="Isi3" class="ui bottom attached tab
segment">
  <div class="ui stackable grid">
    <form action="Unduh" method="post">
      <h4>
        Mengunduh data mulai tanggal&nbsp;
        <input type="date" id="awal"
name="awal" value="2016-01-01">
        &nbsp;hingga tanggal&nbsp;
        <input type="date" id="akhir"
name="akhir" value="<?php echo date("Y-m-d"); ?>">
        &nbsp;
        <button type="submit" class="ui black
button" id="unduh">Unduh</button>
      <br></br>
      </h4>
    </form>
  </div>
  <table class="ui inverted brown table">
    <thead>
      <tr>
        <th>Periode</th>
        <th>Jumlah Data Klaster 1</th>
        <th>Jumlah Data Klaster 2</th>
        <th>Jumlah Data Klaster 3</th>
        <th>Jumlah Data Klaster 4</th>
        <th>Jumlah Data Klaster 5</th>
      </tr>
    </thead>
    <tbody>
      <?php
        for ($i = 0; $i < $jRekap; $i++) {
          echo '<tr>';
          echo '      <td>' . date('d-m-Y',
strtotime(($dataRekap[$i][0]))) . '</td>';
          for ($j = 1; $j < 6; $j++) {
            echo '<td>' . $dataRekap[$i][$j] . '</td>';

```

```

        }
        echo '</tr>';
    }
?>
</tbody>
</table>
</div>

```

```

<script>
$(document).ready(function () {
    $('#DasarMenu').show();
    $('#Tab1').addClass('active teal');
    $('#Isi1').addClass('active teal');
    $('#Isi2').removeClass('active teal');
    $('#Tab1').click(function () {
        $('#Tab2').removeClass('active teal');
        $('#Isi2').removeClass('active teal');
        $('#Tab3').removeClass('active teal');
        $('#Isi3').removeClass('active teal');
        $('#Tab1').addClass('active teal');
        $('#Isi1').addClass('active teal');
    });
    $('#Tab2').click(function () {
        $('#Tab1').removeClass('active teal');
        $('#Isi1').removeClass('active teal');
        $('#Tab3').removeClass('active teal');
        $('#Isi3').removeClass('active teal');
        $('#Tab2').addClass('active teal');
        $('#Isi2').addClass('active teal');
    });
    $('#Tab3').click(function () {
        $('#Tab2').removeClass('active teal');
        $('#Isi2').removeClass('active teal');
    });

```

```

        $('#Tab1').removeClass('active teal');
        $('#Isi1').removeClass('active teal');
        $('#Tab3').addClass('active teal');
        $('#Isi3').addClass('active teal');
    });
    var z = document.getElementById("treeCon");
    z.style.display = 'block';
<?php
for ($i = 0; $i < $jdPeriode; $i++) {
    echo 'z = document.getElementById("treeCon' . $i . '");'
    . 'z.style.display = "none";';
}
?>

    });
</script>
</body>
</html>

```

Source Code PHP, HTML, dan Java Script Untuk Membuat Bagan Balok

```

<style>

/*
.chart rect {
    fill: steelblue;
}
*/

.chart .legend {
    fill: black;
    font: 14px sans-serif;
    text-anchor: start;
    font-size: 12px;
}

.chart text {
    fill: white;

```

```

font: 10px sans-serif;
text-anchor: end;
}

.chart .label {
  fill: black;
  font: 14px sans-serif;
  text-anchor: end;
}

.bar:hover {
  fill: brown;
}

.axis path,
.axis line {
  fill: none;
  stroke: #000;
  shape-rendering: crispEdges;
}

</style>
<svg class="chart"></svg>
<script src="./assets/d3.v3.min.js"></script>
<script>

var data = {
  labels: [
    'Kelembapan', 'Suhu Udara', 'Kualitas Udara',
    'Intensitas Cahaya', 'Getaran Tanah', 'Kecepatan Air',
    'Kecepatan Udara'
  ],
  series: [
    {
      label: 'Data Terbaru',
      values: [<?php

```

```

    for($i=2;$i<9;$i++){
        echo $newData[0][$i];
        if($i<8){
            echo ',';
        }
    }
    ?>]
},
{
    label: "Data Klaster <?php echo $klaster;?>",
    values: [<?php
    for($i=2;$i<9;$i++){
        echo $dataTengah[$klaster-1][$i];
        if($i<8){
            echo ',';
        }
    }
    ?>]
},.]
};

var chartWidth      = 900,
    barHeight       = 20,
    groupHeight     = barHeight * data.series.length,
    gapBetweenGroups = 10,
    spaceForLabels   = 150,
    spaceForLegend   = 150;

var zippedData = [];
for (var i=0; i<data.labels.length; i++) {
    for (var j=0; j<data.series.length; j++) {
        zippedData.push(data.series[j].values[i]);
    }
}

var color = d3.scale.category20();

```



```

var chartHeight = barHeight * zippedData.length +
gapBetweenGroups * data.labels.length;

var x = d3.scale.linear()
  .domain([0, d3.max(zippedData)])
  .range([0, chartWidth]);

var y = d3.scale.linear()
  .range([chartHeight + gapBetweenGroups, 0]);

var yAxis = d3.svg.axis()
  .scale(y)
  .tickFormat("")
  .tickSize(0)
  .orient("left");

var chart = d3.select(".chart")
  .attr("width", spaceForLabels + chartWidth +
spaceForLegend)
  .attr("height", chartHeight);

var bar = chart.selectAll("g")
  .data(zippedData)
  .enter().append("g")
  .attr("transform", function(d, i) {
    return "translate(" + spaceForLabels + "," + (i *
barHeight + gapBetweenGroups * (0.5 +
Math.floor(i/data.series.length))) + ")";
  });

bar.append("rect")
  .attr("fill", function(d,i) { return color(i %
data.series.length); })
  .attr("class", "bar")
  .attr("width", x)
  .attr("height", barHeight - 1);

```

```

bar.append("text")
  .attr("x", function(d) { return x(d) - 3; })
  .attr("y", barHeight / 2)
  .attr("fill", "red")
  .attr("dy", ".35em")
  .text(function(d) { return d; });

bar.append("text")
  .attr("class", "label")
  .attr("x", function(d) { return - 10; })
  .attr("y", groupHeight / 2)
  .attr("dy", ".35em")
  .text(function(d,i) {
    if (i % data.series.length === 0)
      return data.labels[Math.floor(i/data.series.length)];
    else
      return "" });

chart.append("g")
  .attr("class", "y axis")
  .attr("transform", "translate(" + spaceForLabels + ", " +
-gapBetweenGroups/2 + ")")
  .call(yAxis);

var legendRectSize = 18,
    legendSpacing = 4;

var legend = chart.selectAll('.legend')
  .data(data.series)
  .enter()
  .append('g')
  .attr('transform', function (d, i) {
    var height = legendRectSize + legendSpacing;
    var offset = -gapBetweenGroups/2;
    var horz = spaceForLabels + chartWidth + 40 -
legendRectSize;
    var vert = i * height - offset;

```

```
        return 'translate(' + horz + ',' + vert + ')';
    });

    legend.append('rect')
        .attr('width', legendRectSize)
        .attr('height', legendRectSize)
        .style('fill', function (d, i) { return color(i); })
        .style('stroke', function (d, i) { return color(i); });

    legend.append('text')
        .attr('class', 'legend')
        .attr('x', legendRectSize + legendSpacing)
        .attr('y', legendRectSize - legendSpacing)
        .text(function (d) { return d.label; });

</script>
```

Source Code PHP, HTML, dan Java Script Untuk Membuat Bagan Pohon

```

<style>

.node circle {
    fill: #00FF00;
    stroke: steelblue;
    stroke-width: 1.5px;
}

.node {
    font: 10px sans-serif;
}

.link {
    fill: #0000FF;
    stroke: #0000FF;
    stroke-width: 2px;
}

</style>
<div>
    <h5>
        Menampilkan Hierarchical Clustering pada
        periode&nbsp;
        <select class="ui dropdown" id="periode">
            <option value=-1>Minggu Lalu</option>
            <?php
                for ($i = 0; $i < $jdPeriode; $i++) {
                    echo '<option value=' . $i . '>' . date('d-m-Y',
                        strtotime(($dPeriode[$i]))) . '</option>';
                }
            ?>
        </select>
        &nbsp;[Kelembapan ; Suhu Udara ; Kualitas Udara ;
        Intensitas Cahaya ; Getaran tanah ; Kecepatan Air ;
        Kecepatan Udara]
    
```

```

</h5>
<script>
    document.getElementById("periode").onchange =
function () {
    fungsi()
};

function fungsi() {
    var x = document.getElementById("periode");
    var y = x.options[x.selectedIndex].value;
    if (y < 0) {
        var z = document.getElementById("treeCon");
        z.style.display = 'block';
    }
}
</script>

<?php
for ($i = 0; $i < $jdPeriode; $i++) {
    echo 'z = document.getElementById("treeCon" . $i . " ");'
    . 'z.style.display = "none";';
}
?>

<?php
for ($i = 0; $i < $jdPeriode; $i++) {
    echo 'else if(y == ' . $i . '){'
    . 'var z = document.getElementById("treeCon" . $i . " ");'
    . 'z.style.display = "block";'
    . 'z = document.getElementById("treeCon");'
    . 'z.style.display = "none";';
    for ($j = 0; $j < $jdPeriode; $j++) {
        if (!( $i == $j )) {
            echo 'z = document.getElementById("treeCon" . $j .
            " ");'
            . 'z.style.display = "none";';
        }
    }
    echo '}'
}
?>

```

```

    }
    </script>
</div>
<div class="ui sixteen wide column" id="tr">
    <?php
    echo '<div id="treeCon"></div>'
        <script>
            var tData = { name:"",'
            . 'children:[{' . cariAnak($dataNode[0][0], $dataNode,
            $jNode, 0) . '},{
            . cariAnak($dataNode[0][0], $dataNode, $jNode, 1) . '}]'
            . '};</script>
            <script>
                $(function () {
                buildTree("#treeCon",tData);
                });
            </script>';

    for ($i = 0; $i < $jdPeriode; $i++) {
        $jhTree = 0;
        $tTree = array();
        foreach ($hTree[$i] as $row) {
            array_push($tTree, $row);
            $jhTree++;
        }
        echo '<div id="treeCon' . $i . '"></div>'
            <script>
                var tData' . $i . ' = { name:"",'
                . 'children:[{' . cariAnak($hTree[$i][0][0], $tTree,
                $jhTree, 0) . '},{
                . cariAnak($hTree[$i][0][0], $tTree, $jhTree, 1) . '}]'
                . '};</script>
                <script>
                    $(function () {
                    buildTree("#treeCon' . $i . '",tData' . $i . ');
                    });
                </script>';
    }

```

```

    }

    function cariAnak($ot, $dt, $jdt, $j) {
        $temp = "";
        if ($j == 0) {
            for ($i = 0; $i < $jdt; $i++) {
                if ($dt[$i][2] == $ot) {
                    if ($dt[$i][1] > 0) {
                        $temp = 'name: "'
                            . $dt[$i][10] . '"';
                    } else {
                        $temp = 'name: "" ,children:[{' .
cariAnak($dt[$i][0], $dt, $jdt, 0) . '},{ '
                            . cariAnak($dt[$i][0], $dt, $jdt, 1) . '}]';
                    }
                    break;
                }
            }
        } else {
            for ($i = ($jdt - 1); $i > -1; $i--) {
                if ($dt[$i][2] == $ot) {
                    if ($dt[$i][1] > 0) {
                        $temp = 'name: "'
                            . $dt[$i][10] . '"';
                    } else {
                        $temp = 'name: "" ,children:[{' .
cariAnak($dt[$i][0], $dt, $jdt, 0) . '},{ '
                            . cariAnak($dt[$i][0], $dt, $jdt, 1) . '}]';
                    }
                    break;
                }
            }
        }
        return $temp;
    }
    ?>
</div>

```

```

<script>

function visit(parent, visitFn, childrenFn)
{
    if (!parent)
        return;

    visitFn(parent);

    var children = childrenFn(parent);
    if (children) {
        var count = children.length;
        for (var i = 0; i < count; i++) {
            visit(children[i], visitFn, childrenFn);
        }
    }
}

function buildTree(containerName, root, customOptions)
{
    var width = $(containerName).outerWidth() - 10,
        height = <?php echo $jNode * 20; ?>;

    var totalNodes = 0;
    var maxLabelLength = 0;
    visit(root, function (d)
    {
        totalNodes++;
        maxLabelLength = Math.max(d.name.length,
maxLabelLength);
    }, function (d)
    {
        return d.contents && d.contents.length > 0 ?
d.contents : null;
    });

    var cluster = d3.layout.cluster()

```



```

        .size([height, width - 300]);

    var diagonal = d3.svg.diagonal()
        .projection(function (d) {
            return [d.y, d.x];
        });

    var svg = d3.select(containerName)
        .append("svg:svg").attr("width",
width).attr("height", height)
        .append("svg:g")
        .attr("class", "container")
        .attr("transform", "translate(" + maxLabelLength
+ ",0)");

    var nodes = cluster.nodes(root),
        links = cluster.links(nodes);

    var link = svg.selectAll(".link")
        .data(links)
        .enter().append("path")
        .attr("class", "link")
        .attr("d", diagonal);

    var node = svg.selectAll(".node")
        .data(nodes)
        .enter().append("g")
        .attr("class", "node")
        .attr("transform", function (d) {
            return "translate(" + d.y + "," + d.x + ")";
        })

    node.append("circle")
        .attr("r", 4.5);

    node.append("text")

```

```
.attr("dx", function (d) {  
    return d.children ? -8 : 8;  
})  
.attr("dy", 3)  
.style("text-anchor", function (d) {  
    return d.children ? "end" : "start";  
})  
.text(function (d) {  
    return d.name;  
});  
d3.select(self.frameElement).style("height", height +  
"px");  
  
}  
</script>
```